

# **Processors and Processes:**

## **Exploring Computational Design and Landscape Architecture**

By Philip Belesky

A 120-point thesis submitted to the Victoria University of Wellington in partial fulfilment  
of the requirements of the Master of Landscape Architecture (Professional)

Victoria University of Wellington

2013

## *Abstract*

The computer can be a highly efficient drafting table. It can also be much more. Architects can use programming to engage with the computer on its own terms, and in doing so gain a better understanding of complex geometric, structural, or conceptual design scenarios. This ‘computational approach’ to design is increasingly common in architecture, but comparatively rare within landscape architecture. In this thesis I examine how and why landscape architects might employ computational design.

I start by reviewing the work of computational architects and landscape urbanists. I identify that both emphasize diagrammatic and processual strategies as a means to confront complexity and indeterminism within the design process. However, this conceptual overlap masks a technological divergence, as computational tools are presently ill-suited to the needs of landscape architects. Their focus should be shifted away from formal exploration and towards the analysis, simulation, and generation of landscape systems. Doing so would offer landscape architects new forms of representation that would overcome some of the current limitations within their design process.

To test this proposition, I create a series of generative tools, or ‘patterns’, that use computational techniques to model ecological systems. This pattern-based approach introduces a methodology that improves the accessibility and flexibility of computational design. These patterns are applied in tandem with standard computational techniques to create a concept design for a post-industrial landscape. Through this research I identify computation as a powerful tool for designing landscapes. The conceptual and technical methodologies it offers enable landscape architects to better understand and explore open-ended and indeterminate systems. Computation offers a novel opportunity to combine conceptual openness and technical rigour when designing complex landscapes.

## *Acknowledgements*

I would like to thank the academy. In particular, thanks to Jules Moloney and Tane Moleta for their patient and engaging supervision over the course of this thesis. A further thanks is due to both Jan Smitheram and Simon Twose for their insight and encouragement in previous projects.



Contents

1	Introduction	1
1.1	Scope, Structure, and Methodology	1
1.2	Conclusion	2
2	The Architecture of Computation	3
2.1	Computerisation and Computation	3
2.2	Algorithms and Associations	4
2.3	Diagrams and Drawings	5
2.4	A Tale of Two Virtuals	7
2.5	Conclusion	8
3	Landscape Urbanism and its Discontents	9
3.1	McHarg and the Layered Map	9
3.2	Corner and the Virtual Map	12
3.3	Landscape Urbanism and Phased Fields	13
3.4	Architectural Landscape Urbanism	14
3.5	Landscape Architectural Landscape Urbanism	17
3.6	The Root of the Problem	20
3.7	Towards a Landscape Architecture	21
3.8	Conclusion	23
4	Another Pattern Language	25
4.1	Patterns Past and Present	25
4.2	Landscape Patterns	27
4.3	Programmed Patterns	28
4.4	The Patterns	30
4.5	Fields	30
4.6	Flows	31
4.7	Levels	32
4.8	Sections	32
4.9	Seeds	33
4.10	Conclusion	34

5	Case Study: Seaview	39
5.1	The Brief	39
5.2	Mapping Analysis	43
5.3	The Initial Design Proposal	44
5.4	Subsequent Design Proposal	63
5.5	Evaluation of Patterns	79
5.6	Limitations in Programming	80
5.7	Limitations in Representation	81
5.8	Conclusion	82
6	Exegesis	83
6.1	The Conceptual Investigation	83
6.2	The Design Investigation	84
6.3	Conclusions	85
7	Appendix A: Process Philosophy and Real Virtualities	87
7.1	The Problem of Universals	87
7.2	Intensive Phenomena	89
7.3	The Manifold	89
7.4	The Attractor	90
7.5	The Virtual Multiplicity	91
7.6	Conclusion	91
8	Appendix B: The History of Seaview	93
8.1	1000: A Prehistory	93
8.2	1250: A Settlement	94
8.3	1800: A Change	94
8.4	1855: An Uplift	95
8.5	1900–Present: Development	95
8.6	Conclusion	96
9	Endnotes	97
10	Bibliography	107
11	Illustration Credits	123

# <sup>1</sup> *Introduction*

Computational design techniques are often used to help design buildings, but rarely used to help design landscapes. In this thesis I explore how and why computation should be more prevalent in landscape architecture.

I do this by questioning the mindset and toolset we use to design landscapes. I demonstrate that a computational design process can enhance our understanding of the environment, and our designs upon it. I speculate on the future practices of my discipline by asking: “How can computational design improve landscape architecture?”

## 1.1 **Scope, Structure, and Methodology**

To answer this question, I follow a design research strategy. Theoretical ideas and their implementations are explored in tandem throughout the design process.

The first section of this thesis addresses questions of strategy. Very little literature explicitly investigates the intersection of computational design and landscape architecture, so the scope of my theoretical research is wide. Across two chapters I examine two fields: computational architecture, and landscape urbanism. While discussing both fields, I establish critiques that are later synthesised to establish a vision of, and a justification for, a landscape architecture that has embraced computation.

The second section of this thesis addresses questions of tactics. Across two chapters I examine the use of computation design techniques, as both a general and an *in situ* practice through two case studies.

The initial study investigates the utility of computational tools for landscape architectural practice in general. I do this by programming a series of ‘design patterns’ that each investigate a particular problem domain in landscape architecture, such as hydrology, grading, or planting. Each pattern provides computational techniques that enhance the design process when examining a particular problem domain.

The latter study investigates the utility of computational tools in developing a particular design. Here I use the patterns to help create two concept designs for a specific client and site. I then evaluate the value of my design patterns and other computational techniques within landscape architectural design.

In the exegesis I review the research as a whole: summarising where theory and practice meet, where they diverge, and what future opportunities this suggests.

## 1.2 **Conclusion**

Computational design is now common in architecture, interior architecture, and engineering. There is a growing need for designers to understand these tools, particularly when working collaboratively.<sup>1</sup> Moreover, computation has enabled designs that were previously impractical, if not impossible. It is significant that landscape architecture remains largely unengaged with the potential of computational design.

The year-long scope of this thesis, and the emerging nature of its subject, mean that my findings cannot claim to be comprehensive or conclusive. Yet, now is the best time for such an investigation. If landscape architecture is to adopt computation, the challenges and opportunities identified in my research, and the research of others, are most valuable during these formative stages.

## 2 The Architecture of Computation

Digital morphogenesis, digital architecture, cybernetic architecture, topological architecture, evolutionary architecture, algorithmic design, computational design, emergent design, generative design, and parametric design.

These are the seeds that computation has sown into the field of architecture. While each differs from the others in its ideas and execution, they share a common methodology that sets them apart from traditional design methods. This chapter introduces this methodology by discussing the distinction between computational and computerised design methods. From there I discuss why computational design methods can be considered a novel form of diagrammatic design strategies. I finish by explaining the relevance of Deleuze to computational design.

### 2.1 Computerisation and Computation

*“While computation is the procedure of calculating, i.e. determining something by mathematical or logical methods, computerization is the act of entering, processing, or storing information in a computer or a computer system. Computerization is about automation, mechanization, digitization, and conversion. Generally, it involves the digitization of entities or processes that are preconceived, predetermined, and well defined. In contrast, computation is about the exploration of indeterminate, vague, unclear, and often ill-defined processes; because of its exploratory nature, computation aims at emulating or extending the human intellect.”<sup>2</sup>*

Architects have been using computers as design tools since the 1960s. These early forays involved manually creating programs that would create, store and visualise geometric data.<sup>3</sup> Designing and programming went hand in hand.

Much has changed since then. Over the course of the 1990s, Computer Aided Design (CAD) software became an indispensable part of mainstream architectural practice.<sup>4</sup> CAD software was adopted because it was accessible: architects could create geometry by using simple tools and commands — LINE, CURVE, CIRCLE, MOVE, ROTATE — rather than by programming. The ascendancy of CAD software over bespoke programs changed the nature of the computer as a design tool: it became a tool for *computerisation* rather than a tool for *computation*.

Because the interface of CAD software mimics conventional drafting techniques, it imposes limits on the design process. Such standardised tools restrict the range of geometric operations available,<sup>5</sup> heavily favouring entrenched architectural forms and methods that marginalise alternative practices because conventional paradigms of architectural design are encoded into the operation of the software.<sup>6,7</sup> Because “architects tend to draw what they can build, and build what they can draw”<sup>8</sup> the adoption of CAD software significantly limited the range of architectural expression. As Timothy Lenoir



FIGURE 1. A demo of the Sketchpad system in 1963.  
“It’s like a picture within a picture!”  
“Right, it’s real nightmare material.”

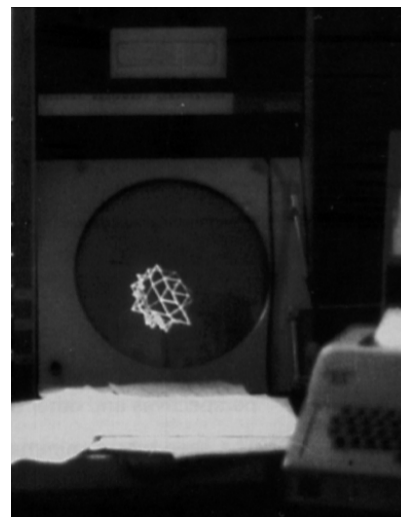


FIGURE 2. John Frazer’s reptile structural system, running on a computer at the Cambridge University mathematical laboratory in 1963.  
“I had to write the generative instructions in Atlas Titan machine code, and if you don’t know what that is then you don’t know how lucky you are!”

and Casey Alt note, “Coloring outside the lines is not only stylistically unadvisable in CAD programs; it is technologically impossible because there is no tool to allow it.”<sup>9</sup> Even when used for documentation and presentation, rather than design, CAD software encourages projects to be refactored to fit within its limits.<sup>10</sup> The computer becomes a black box. It loses its ability to generate specialised forms on a project-by-project basis, and is used to represent preconceived forms rather than to design new forms.<sup>11,12</sup>

CAD’s reliance on standardised tools is evidenced in the built environment. Over the course of the 1990s, architecture increasingly employed complex curvatures.<sup>13</sup> The popularity of such forms can be directly traced to new CAD software that allowed for the manipulation of spline curves and surfaces.<sup>14</sup> Although these tools could produce highly complex results, they were still predominantly used as methods of computerisation in that they were a means to represent or optimise a pre-imagined design. This is demonstrated in the design process of Frank Gehry, where his buildings are conceived through physical models that are then reproduced digitally to test aesthetic and structural outcomes. Many architects use CAD software as a kind of improvised sculpting environment where form is manipulated according to the successive musings of the designer.

Over the last decade, architects have increasingly explored computation as a design tool.<sup>15</sup> By once more designing buildings using code they escape the conforming influence of codified procedures. In doing so they create their own tools and thus enter a higher form of craftsmanship where the constraining influence of CAD software can be overcome.

### 2.2 Algorithms and Associations

Algorithms are logical procedures that address a problem through a finite number of steps. At the most basic level, everything that a computer does is an interaction between algorithms and information, between code and data.<sup>16</sup> In order to manipulate data for architectural ends, algorithms must typically produce geometric data. For example, to create a perspective view of a list of geometric data, an algorithm is run whereby each of the Euclidean co-ordinates is ordered, arranged to face the viewport, and then positioned linearly to simulate the diminishing vanishing point of perspective projection.

The key feature of computational design is that it is algorithm-driven. Designers create geometry by engaging with the computer’s internal logic, rather than by directly manipulating a graphic representation. A designer’s intent is translated directly into algorithms, escaping the predefined limits of standardised CAD tools.

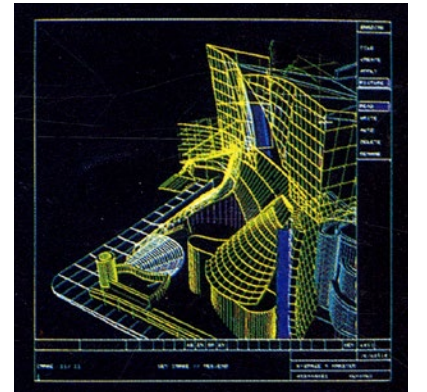


FIGURE 3. There is more than one way to skin a building: an early design for the Walt Disney Concert Hall in CATIA.

However, designing with algorithms comes at a cost. It requires the designer to work with the formal languages of computer code, rather than drawing and understanding geometric relationships intuitively.<sup>17</sup> Having to explicitly represent the assumptions that define a design requires that these assumptions be very well understood.<sup>18,19</sup> This limitation is also computational design's greatest strength. Beyond the ability of the computer to quickly perform complex geometric calculations, the key benefit of designing using algorithms is that it requires articulating design intent in the language of mathematics and programming.<sup>20</sup> This medium is a highly flexible one. The ease of creating, reconfiguring, and reverting geometric relationships acts as a catalyst for conceptual development because the designer does not need to manually rebuild a detailed model after changing the underlying design logic.<sup>21</sup>

This new design process creates both a change in designed outcomes and also a change in how the architectural project is understood. A design is no longer an object, but a system defined by a series of interrelated systems, whether they are geometric, material, technological, or programmatic.<sup>22,23,24</sup> It entails that architecture be defined from first principles, creating a “method of constituting the architectural project in a long sequence of relationships from the first conceptual hypotheses to the driving of the machines that prefabricate the components that will be assembled on site.”<sup>25</sup> A design exists as a dynamic ecosystem of algorithmic relationships: a collection of rules rather than a collection of shapes. The designer is then a “controller of processes”<sup>26</sup> who creates a rigorous system of interrelated rules.<sup>27,28</sup> This is the first reason why computational design is radical.

## 2.3 Diagrams and Drawings

Present throughout the history of architecture, diagrammatic methods have recently become a favoured methodology within the design process.<sup>29</sup> Propelled by post-critical discourse, the increased use of the diagram in theory and practice serves to highlight the distinction between abstract and mimetic modes of graphical representation.

The architectural design process typically employs mimetic representations that translate architectural thought into pictorial graphics. These methods typically rely on conceiving of architecture as eidetic scenes: mental conceptions of space, form and experience that while pictorial, are also “equally acoustic, tactile, cognitive or intuitive.”<sup>30</sup> Whether through the palimpsest of sketching, or the codes of drafting, the traditional role of drawing has been to create graphic representations that function as proxies for our eidetic imagination. In rarefying intention into image, we are able to test and revise their designs against the realities of form and function. In this way the drawing seeks to “function between the architect (idea) and the building (the realisation of that idea).”<sup>31</sup>

In contrast to mimetic representations are non-pictorial methods of representation, such as the diagram. Diagrams operate outside the confines of pictorial drawing codes, and so are free to detail abstract ideas rather than precise forms. While they may incorporate formal elements, they primarily examine *conceptual* compositions rather than *geometric* compositions. As the

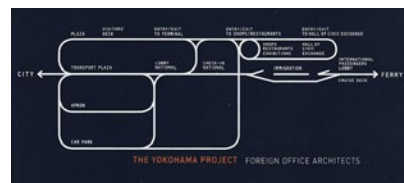


FIGURE 4. Quite the parti: FOA's circulation diagram for the Yokohama Port Terminal.

sketch enables the direct interrogation of formal concerns, the diagram enables the direct interrogation of conceptual concerns. This is what makes it such a crucial tool in articulating and developing the design process. By making conceptual intents explicit, architects can interrogate their ideas in the same way they interrogate their forms: by representing them visually. Thus diagrams not only image the intent of the architect, but become actively involved in developing that intent. Following Stan Allen, the diagram is “an abstract means of thinking about organisation”: a design space that is conceptual-organisational, rather than pictorial-eidetic. A configuration is crafted that is “not a thing in itself, but a description of potential relationships among elements.” It does not resemble what it produces.

In contrast, operating CAD software requires manipulating a 3D visualisation of architectural form. Changes are seen immediately, and the resulting forms can be further modified according to the whims of the designer. Given this methodology, it is clear that most CAD software operates in a pictorial mode because it relies on the production and manipulation of graphic representations. However, more so than traditional methods of drawing, CAD software heightens this focus on the pictorial. It effortlessly produces accurate perspectives, enabling forms to be visualised and judged with unparalleled ease. As a result, all design concerns that cannot be presented as architectural forms are marginalised, and the need for architects to conceive of their design as fully eidetic images is suppressed. As noted by Eugenia Victoria Ellis, the “imaginative construction of the mind’s eye is taken care of automatically by the computer ... Images that once were fabricated within the mind are outside of ourselves, rotating freely within the cathode-ray tube.”<sup>32</sup> Thus, in adopting CAD software the architectural design process has come to rely ever more heavily on pictorial methods of representation, privileging the digital perspective to the extent that it has begun to displace the architect’s own eidetic imagination. What was once a feedback loop between an eidetic imagination and pictorial representation has become a closed loop where both imagination and representation are confined to the screen.

When using computational methods the architect must express their design intent using the textual medium of computer code. This process is analogous to diagramming, as both computational design techniques and drawn diagrams articulate a design by defining relationships between ideas. Following Pia Ednie-Brown, “programming involves diagrammatic thinking, operating through notating and mapping out the interplays of relations.”<sup>33</sup>

Yet there is a crucial difference between computational design and the traditional use of the diagram. Because the 'digital diagrams' of programming operate within the computer, there is the option for a visual representation of geometric form to be generated alongside the programming process.<sup>34</sup> In contrast, drawn diagrams can only engage with formal concerns under highly abstracted conditions where form is reduced to simple shapes or iconographies. The digital diagram escapes such reductive abstraction because it can concurrently produce a full pictorial visualisation of the underlying design logic. Here the computer offers a stereovision where the defining-formula



can be juxtaposed against the defined-geometry. In this way computational design creates a new form of diagram, one that employs non-formal methods to articulate conceptual intent while simultaneously resolving it into detailed form.

It is this ability to render diagrammatic methods into form that further distinguishes computational design from the traditional methods of diagram-based design. When using computational methodologies the relationships between architectural elements define conceptual *and* formal intent — the digital diagram can synthesise both mimetic and non-mimetic representations of architectural ideas. Moreover, because the formal results of computational operations are often highly complex, their results cannot be precisely visualised by the architect. Instead the exact resolution of a diagrammatic operation can only be seen by enacting the operation and considering the result.<sup>35,36</sup> Unlike a traditional mode of pictorial representation, the architect does not seek to represent a formal concept. Instead computational procedures are used to discover and guide form, tweaking and re-configuring the programmed parameters and relations to discover and refine forms:

*“We see a development in the very nature of the architect from the demiurgic ‘form-giver’ to the architect as the controller of generative processes, where the final appearance is a product not of the architect’s imagination alone, but of the generative capacities of computer programs.”*<sup>37</sup>

Computational design bridges the gap between diagrammatic and pictorial modes of design, having created a synthesis in which design intent is enacted through diagramming at the same time as the results are evaluated through pictorial representations. This juxtaposition of diagrammatic and pictorial methodologies within the digital environment has enabled a novel method of design where formal and conceptual elements are intertwined in flux. Unlike the prior paradigms of design — in which the pictorial and diagrammatic were disparate — the design process becomes about resolving the ambiguity and tension between these two modes. This is the second reason why computational design techniques are radical.

## 2.4 A Tale of Two Virtuals

The work of Deleuze has been the dominant mode of theorising digital architecture since its re-emergence during the 1990s. This has come in two waves.

Until recently, the discussion of Deleuze in architecture was largely focused on particular concepts — such as the fold or smooth space — and was largely uninterested in the more fundamental aspects of his philosophy. Aesthetic appropriations of these concepts quickly became clichéd.<sup>38</sup>

Rising in tandem with architects rediscovering computational design techniques has been a complementary emphasis on the ontological concepts developed by Deleuze. This should not be surprising. Both are primarily concerned with a reification of the virtual, in that both the virtual of Deleuze and the — virtual of computer systems operate by defining rules that realise

processes. Neither the multiplicity, nor the computational, represent form, but rather represent the intrinsic instructions that generate form.<sup>39,40,41</sup> Thus, the procedural operations of a computer can thus be considered analogous to the phase space of the virtual multiplicity because each defines abstract features — either the logic of algorithms or through the features of manifolds — that guide actualisation. In neither case does the product resemble the process.<sup>42,43</sup> Using computational techniques we can understand and develop a design as a population of possibilities within the degrees of freedom granted by an assembly of evolving rules.<sup>44</sup> Our design process instrumentalises Deleuze’s ontology.

Computational tools offer a fountainhead of possibility. While they require technical skill to operate, the key challenge in using computational design is in adopting computational thinking into design process.<sup>45,46,47</sup> Designers have yet to “incorporate the architecture of computation into the computation of architecture.”<sup>48</sup> An understanding of this affinity between Deleuze and the digital enhances the design process by providing the means and impetus for designers to more consciously and productively engage with computation. In particular, a more thorough application of Deleuzian thought could help expand the use of computational design beyond the formal extravagance that it is commonly criticised for. By collapsing the distinction between forms and process, the ontology of Deleuze suggests that the same rules that create architectural geometry can be used to better engage with the performative aspects of a design’s contexts and contents.<sup>49,50</sup>

## 2.5 Conclusion

This chapter has outlined the conceptual distinctions between computation and computerisation, and detailed how this affects the design process of architects. The computational design process uses programmed algorithms to define diagrammatic rules, working in conjunction with the automated presentation of form.

A key constraint is the need to more rigorously apply computational thinking to the design process itself. In order to bridge this gap, I suggest that the work of Deleuze offers a conceptual framework that can guide the use of computational tools. This potential derives from the fact that Deleuze’s ontology has a distinct affinity to the medium of programming: both mandate an understanding of the world as composed of generative rules.

## 3

questioning.”<sup>51</sup>

*soils, plants, and animals.*"<sup>52</sup>

Ecology radically altered humanity's conception of the environment. It is significant that landscape architecture is one of the few professions to use ecological science in both an analytic and generative role. New forms of representation were critical in the initial, and successive, integrations of ecology into the design process.

This chapter summarises this development in the recent history of landscape architecture. I examine the shifts and shortfalls in mapping practices and ecological thought that lead to the contemporary practice of 'landscape urbanism'. Landscape urbanism is then criticised, and the use of computational design strategies suggested as means of addressing these critiques.

## 3

Although landscape architecture has always incorporated ecological *concerns*, Ian McHarg was the prime mover in introducing ecological *science* into the discipline.<sup>53</sup> Through the lens of the ‘ecological perspective’ landscape architects would come to better understand their sites as complex systems of biotic and physical components. Doing so required the profession to appropriate parts of the scientific method, and to aspire to problems more complex than the design of parks and gardens.<sup>54,55</sup> Following McHarg, the landscape architect would take their place at the nexus of environment planning and environmental design.

The key value of McHarg's methods was his advocacy for an interdisciplinary methodology in mapping and surveying. Following ecological science's broad scope, McHarg proposed that surveys should be undertaken by a diverse team of scientists under the leadership of a landscape architect. Each scientist would analyse the landscape in terms of their specialisation, reporting and recording the results. After extracting each type of information and mapping it on to the landscape, the knowledge from each scientific field could begin to be understood in relation to spatial conditions. By overlapping these slices of data using transparent materials, each layer acted as a 'thin film of information' that would be overlaid into a thick 'layer-cake'.<sup>56</sup> The landscape could then be read as a palimpsest of interacting systems; as a detailed ecological understanding

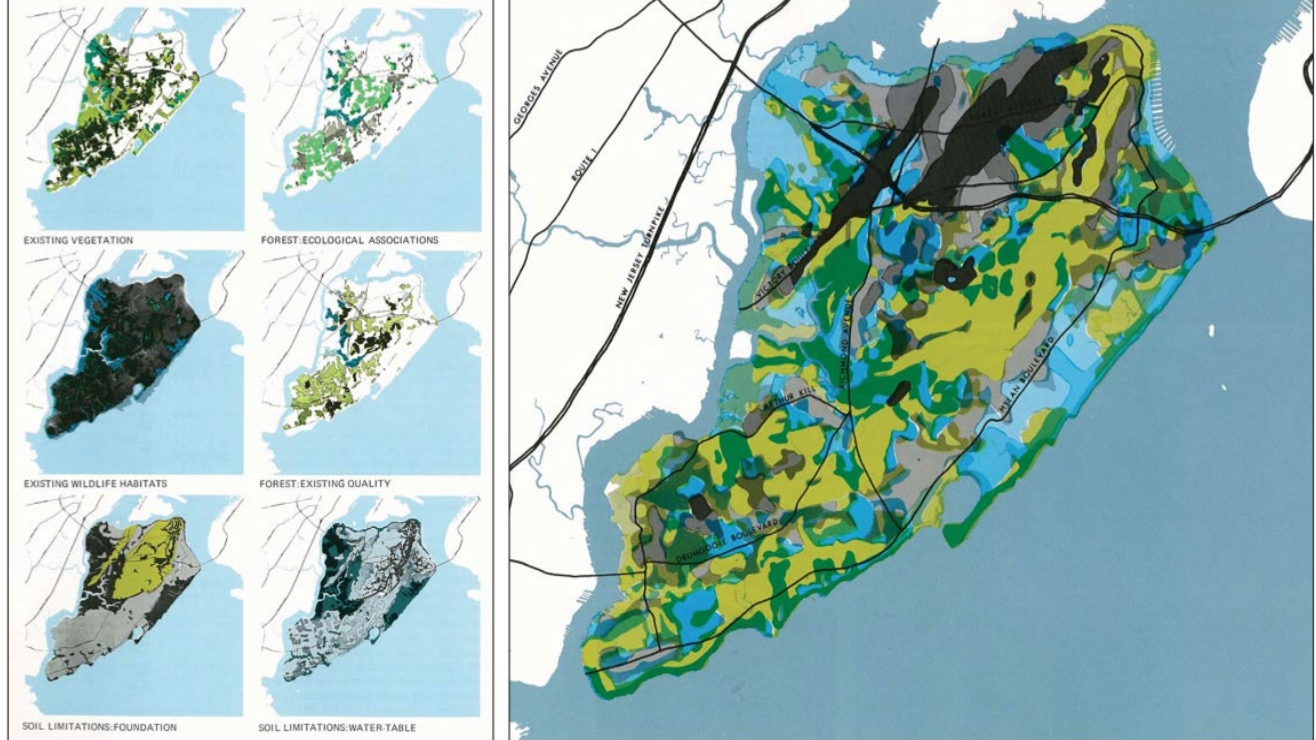


FIGURE 5. Ian McHarg's 'layer-cake' plans for Staten Island, 1968.

[illegible]

FIGURE 6. One of Ian McHarg's 'suitability matrices.'



of the connections between complex and multi-scalar environmental systems. After this knowledge was assembled the landscape architect would have an unprecedented understanding of the functional systems present within the site.

The use of overlays within mapping was not novel, but McHarg’s use of ecology as the organising framework for this layering was. In transforming scientific understanding into a visual artefact, a complete understanding of how an environment operates could be approached and leveraged within the design process as a ‘suitability analysis’ where the compiled data could be evaluated against the requirements of the brief and locations assigned varying scales of utility. For example, the fertility of soils could be assigned positive values according to its ability to support agriculture, while the height of the water table might be assigned negative values according to its inability to support built settlements. By interpreting the gathered spatial information against these criteria, the resulting composite would reveal areas of suitability within the landscape that were especially compatible with the brief.

McHarg’s methods were rapidly adopted into mainstream landscape architecture;<sup>57</sup> have been institutionalised in planning and environmental assessment in much of the world;<sup>58</sup> and provided much of the inspiration for the layer-based approach of CAD and GIS systems.<sup>59,60</sup> They have also been widely criticised, setting the stage for a re-evaluation of ecology and mapping in landscape architecture.

Many criticisms of McHarg attack the supposed strength of the ‘layer cake’ method — its use of the scientific mode of survey and analysis — as forcing the design process to assume a positivist framework wherein all phenomena must be understood objectively. Because each layer of information is just a series of numbers charted to locations, the resulting map ends up as a birds-eye view upon a strata of data. This perspective ignores subjective effects, such as aesthetic appearance and cultural importance, because it only recognises quantifiable information. Although McHarg’s maps would document the location of historic sites, socio-economic statistics, and scenic values, their method of representation presumes that this cultural information can be directly compared to scientific surveys. Yet this requires passing culture through a reductive Cartesian sieve that erases the complexities of subjective experience that produced the significance in the first place.

This was particularly problematic given that McHarg’s framework was designed to inflexible and deterministic.<sup>61</sup> He believed “that anyone would reach the same conclusions” and that the “engineer, architect, landscape architecture, developer, and the client himself were bound by the data and the method.”<sup>62</sup> Although his method was titled “a theory of creative fitting,” his design process was largely incompatible with creative expression, aspiring instead to become a purely scientific undertaking.<sup>63,64</sup>

One of McHarg’s lasting contributions was his definition of “nature as a process.”<sup>65</sup> While this was an influential piece of thinking, such a dynamic and temporal concept of the environment is not present in his maps because they only ever depicted an analysis of the present landscape. Despite advancing

the notion that “morphology is a superficial expression of the process examined,”<sup>66</sup> the ecological perspective is never fully actualised within his representational techniques.<sup>67</sup> There is a clear gap between the theoretical implications of his work and their realisation in his designs.<sup>68</sup>

### 3.2 Corner and the Virtual Map

In the decades following McHarg’s popularity, the landscape increasingly became a medium for formal expression, as demonstrated by works such as the Vietnam Veterans Memorial or the land art of Robert Smithson. These expressive designs contrast heavily with the rational approach of McHarg, occupying a niche that had emerged when landscape architects became “analysts rather than creators”<sup>69</sup> and the “conscious making of form and space in the landscape subsequently came to a screeching halt.”<sup>70</sup> As a response to this deficiency, a newer generation of landscape architects pushed for the scientific and aesthetic uses of ecology to be united. Of these, James Corner is the most emblematic, having noted that the “possibilities for a vibrant exchange between ecology, creativity and the design of landscape have barely been recognized beyond mechanical and prescriptive methods.”<sup>71</sup> The positivism of McHarg’s methods was identified as a “tyranny.”<sup>72</sup>

It is no surprise that Corner made extensive references to the work of Gilles Deleuze in his writing. A philosophy of generative processes was apt to advance landscape architecture towards a more integrated approach when working with both natural and cultural systems, where landscapes would be measured in terms of the “various hidden forces that underlie the workings of a given place”<sup>73</sup> that serve to define a landscape as a “complex and dynamic imbroglio of social and natural processes.”<sup>74</sup> To actualise this perspective, Corner developed a new methodology of mapping that could re-introduce these aspects into the design process by rendering them coherent through a process of visualisation and interrelation:

- The first step is to create a ‘field’ by consciously defining the cartographic scale, scope, and symbols that would be used. This defines the apriori features of the map itself. The graphic conventions of mapping were given such special attention because they were recognised as having been “codified, naturalized and taken for granted.”<sup>75</sup>
- Once the features of the field were constructed, an ‘extraction’ process takes place by which information from a given milieu is identified and marked according to the established conventions.<sup>76</sup>
- The final step is a ‘plotting’ where the relationships between each of the extracts are examined. Here the mapmaker reveals and constructs the creative possibilities presented by their choice of field and extract.<sup>77</sup>

This method was used to create the maps in *Taking Measure Across the American Landscape* that identify and subvert the conventions of agricultural surveys to represent social conditions. These marks were then collaged with both traditional maps and photographs to plot the connections between these phenomena and the landscape.



FIGURE 7. Robert Smithson’s “Broken Circle.”



FIGURE 8. The Vietnam Veteran’s Memorial.

Corner’s advocacy of a more critical map-making process is one of the most important contributions to contemporary landscape architectural theory.<sup>78</sup> In a stark contrast to McHarg’s map-making, Corner does not dictate the exact methods of mapping, but rather a process that enables designers to be more critical of the mapping process itself. His methods expose the true subjectivity underlying the sense of authority a map confers, demonstrating that no technique was “neutral, passive, or without consequence.”<sup>79,80</sup>

Corner’s methods draw heavily from the concept of mapping developed within Deleuze’s work. To Deleuze, mapping is the opposite of tracing, a binary that mirrors his ontological distinction between virtual and actual. As applied to landscape architecture, a tracing is a representation of the instantaneous, actual state of a landscape. A mapping is a more productive investigation, one that involves “actualizing within its virtual spaces new territories and prospects out of pervasive yet dormant conditions.”<sup>81</sup> As mentioned in the earlier discussion on McHarg, this static and discrete quantification of an environment obscures the dynamic processes that actually generate such phenomena.

While Corner’s re-conceptualisation of mapping identifies the virtual procedures that define mapping itself, the process of mapping in a Deleuzian sense is not manifested in the maps he creates for *Taking Measure Across the American Landscape*. While novel for their representation of the interactions between culture, industry, and the landscape, these systems are only ever presented in terms of their manifestations as *actual* conditions.<sup>82</sup> There is a missed opportunity to represent the more virtual conditions that explain *why* these phenomena have come about. Corner’s maps largely operate as enlightened tracings rather than as true explorations of virtual conditions.

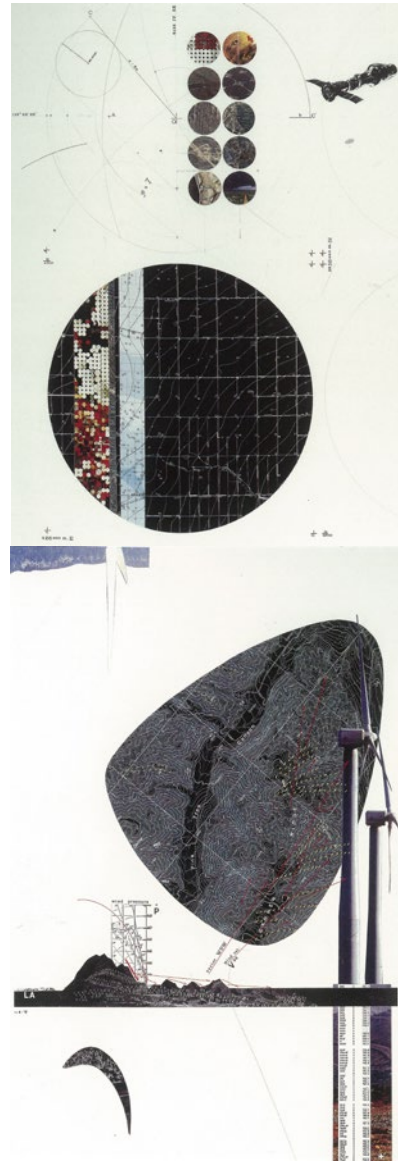


FIGURE 9. Collages from *Taking Measure Across the American Landscape*.

### 3.3 Landscape Urbanism and Phased Fields

Landscape Urbanism is just one of many ‘urbanisms’ to emerge in recent decades. Like its siblings, landscape urbanism was conceived as it became clear that the modernist city was terminally ill.<sup>83</sup> The modernist master plan assumed that cities would grow in an incremental and predictable fashion; a determinism that proved particularly problematic as globalisation saw increasingly rapid and volatile growth patterns — compare the Detroit, Shanghai, or Dubai of today to their state at the turn of millennium. Moreover, modernism’s emphasis on large-scale single-purpose infrastructures — stormwater channels, motorway systems, and the like — led to projects that were all too often ineffective, costly, detrimental to their contexts,<sup>84</sup> and increasingly inflexible in the face of new challenges such as climate change.<sup>85</sup>

Landscape urbanism aims to right these wrongs by emphasising the role of landscapes in urban environments. Landscapes are posited as the correct ‘lens’ by which to view the design and planning of cities because they are processual medium that is “capable of responding to temporal change, transformation, adaptation and succession (and) therefore uniquely suited to the open endedness, indeterminacy and change demanded by contemporary urban conditions.”<sup>86</sup> This capacity for change contrasts to the object fixation of traditional architectural and planning practices where the city is viewed as

collections of static objects.<sup>87</sup> In contrast, the landscape urbanist perspective sees the city as a coagulation of natural and urban flows, its composition shifting according to a multitude of processes interacting over time.<sup>88</sup> If the Futurist’s city was a radiation of cars, trains, and crowds,<sup>89</sup> the landscape urbanist’s city is rhizome for the flows and fluxes of water, waste, weather, species, senses, spaces, and events.

Seeing the city from this perspective expands the concept of a landscape beyond the dichotomous models of traditional planning — *the park outside the city* — and modernist planning — *the city inside the park*. Instead urban areas, suburbs, industrial areas, et al, exist and operate under the same terms — as landscapes.

In further discussing the landscape urbanism, I will distinguish between two broad schools of thought: one more architectural, the other more landscape architectural. These two camps overlap to a large extent, but the distinction is useful in discussing the different kinds of criticism brought against landscape urbanism.

### 3.4 Architectural Landscape Urbanism

The first camp of landscape urbanism emerges from, and focuses upon, architecture. Here the 1983 competition for Parc de la Villette is key.<sup>90</sup> Both the winning and runner-up entries were designed by architects. Each was noted for eschewing a traditional approach to designing large scale parks in favour of a new landscape through the use of diagrammatic operations:

- Tschumi’s design imagines three individual layers in plan that each represent a different type of form to be constructed. These were the points (follies), the lines (paths) and the surfaces (fields). By vertically superimposing these layers the park becomes a complex mix of spaces, forms, and programmes.
- Koolhaas’ design imagines a series of long and narrow strips, each representing a distinct programme and spatial type. These layers are then horizontally juxtaposed in plan, creating typologies and programmes that are longitudinally consistent, but transversally contrasting.

Both entries make use of simple diagrammatic logics operating in plan to create a clear, structured, and programme-driven design.<sup>91</sup> This strategy circulated throughout architectural theory and practice during the 1990s, forming a part of the avant-garde’s interest in open-endedness and indeterminacy amongst other post-modern and post-structuralist concerns. Architects such as OMA, MVRDV, and FOA demonstrate this practice in architectural design.<sup>92,93</sup> Texts such as Koolhaas’ *Generic Cities* and *Junkspace*, Stan Allen’s “From Object to Field,” Bernard Tschumi’s explorations of the Event, and Kenneth Frampton’s discussions of critical regionalism advanced these concepts in theoretical discourse.<sup>94</sup>

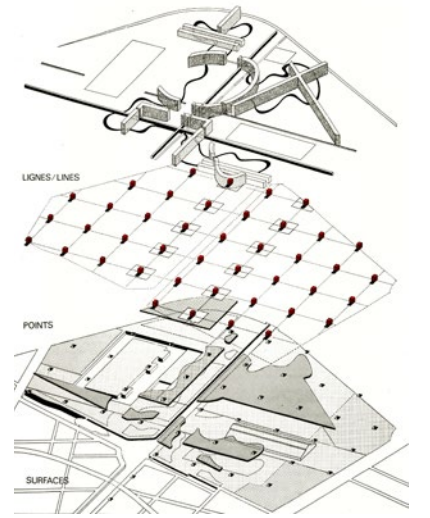


FIGURE 10. Bernard Tschumi’s design for the Parc de la Villette.

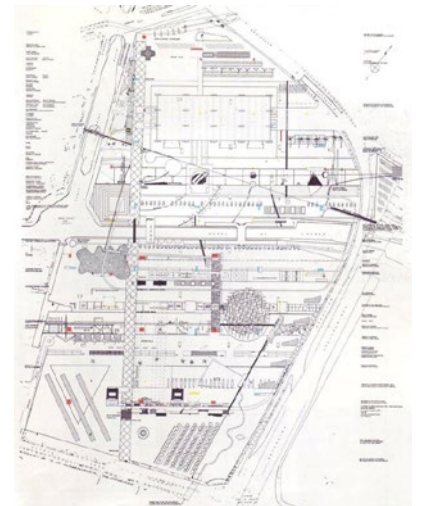


FIGURE 11. Rem Koolhaas’ design for the Parc de la Villette.



This set the stage for the Landscape Urbanism conference that was held in 1997. Three years after the conference, the Architectural Association (AA) started its own landscape urbanism programme, and three years after that published *Landscape Urbanism: A Manual for the Machinic Landscape*. This text, and the work of the AA's programme best exemplify this more 'architectural' strain of landscape urbanism.<sup>95,96</sup> It can be distinguished by a number of characteristics:

- There is less of an emphasis on landscape architectural theory and ecological science, and more of an emphasis on contemporary architectural theory.<sup>97</sup> It has been criticised as a transdisciplinary, rather than interdisciplinary pursuit, in that it selectively appropriates ideas from landscape architecture into architecture.<sup>98,99</sup>
- The design process is more distinctly architectural, focusing on how architectural design strategies can be applied horizontally rather than vertically.<sup>100,101</sup> This leads to designs in which “buildings are landscaped and landscape becomes architectural”, replacing the traditional figure/ground configuration with a more topographic configuration.<sup>102</sup>
- The design process makes heavy use of computer modeling, along with the occasional use of computational techniques to analyse and design in response to landscape conditions.
- Designs often take the form as 'continuous surfaces' that operate as interconnected superstructures of architectonic forms.<sup>103,104</sup> The emphasis on 'infrastructure' promoted by landscape urbanism is usually applied through formal interventions.<sup>105</sup>
- The selection of sites encompasses a range of environments wider than just developed cities, post-industrial zones, and large parks. Student work — and to a certain extent the work of several practices — regularly examine sites within countries such as Mexico, Sri Lanka, the United Arab Emirates and China.<sup>106</sup>
- There is a generally greater engagement with theory, particularly that of Deleuze, and how design can act as its praxis.<sup>107</sup>

In this way, the practices of AA-centric landscape urbanism exist as an updated, but still flawed, evolution of the strategies introduced at the Parc de la Villette competition. Whereas Tschumi and Koolhaas' entries emphasized diagrammatic clarity, they did so at the cost of imposing tabula-rasa,<sup>108</sup> negating existing site conditions and processes while imposing designs that celebrate programme but do little to express or develop the landscape itself.<sup>109</sup>

The work of the AA similarly employs diagrammatic strategies to generate formal interventions. That said, their diagrams build upon a more rigorous analysis and understanding of the existing landscape that leads to an adaptive complexity whereby their designs tend to bend and grow according to a contextual application of an overarching logic. However, the architectural nature of these forms leaves these design solutions largely static and inflexible; an imposition that imposes a new equilibrium upon the landscape.<sup>110</sup> This acts to neuter both the ability of the existing landscape systems to change, and the ability of the design itself to change in response to future problems.<sup>111,112</sup> The design enacts a short-term reconfiguration, rather than a long-term remediation.<sup>113,114</sup>



FIGURE 12. One of the few examples of a built project that typified the AA-style landscape urbanism: Groundlab Landscape Urbanism's design for the "Flowing Gardens" of the Xian International Horticultural Fair.



3.5 Landscape Architectural Landscape Urbanism

At the other end of the scale is landscape urbanism as advanced by three students of Ian McHarg: James Corner, Charles Waldheim, and Chris Reed; several prominent practices such as Field Operations, STOSS, and WEST8; *The Landscape Urbanism Reader* publication; and projects such as Fresh Kills Park.

To distinguish this group from the ‘architectural’ strain of landscape urbanism is to draw a false dichotomy — the projects and texts mentioned above were heavily influential here and vice versa. Yet there are distinguishing features to the origins and practices of this group that are marginalised by the AA’s practices of landscape urbanism.

The most notable feature is a more thorough recognition and critique of the history of landscape architecture and planning since McHarg. As noted earlier, there was a push during the 1990s by Corner and others to use ecology to create more expressive designs. Many landscape urbanist techniques for surveys, analysis, and design trace directly back to McHarg, but have updated his process by expanding the scope of what should be considered during the design process.<sup>115</sup> The close relationship between ecology and landscape architecture since McHarg explains many of these developments.

Starting in the 1980s, and continuing to the present, ecology has increasingly focused upon how human populations and spatial patterns interact with ecosystems at a variety of scales. Richard Forman and Michel Godron’s *Landscape Ecology* pioneered this approach, enabling ecology to be used as a tool for understanding and improving land-use patterns.<sup>116</sup> This paved the way for further works, and the foundation of ‘urban ecology’ that popularised an understanding of cities as complex hybrids between natural and artificial ecosystems.<sup>117</sup>

In the face of this development, McHarg’s conservationist rhetoric, which sought to separate inhabitation and nature was found increasingly untenable.<sup>118</sup> The emergence of urban ecology demonstrated that even supposedly isolated ecosystems are heavily influenced by the effects of human inhabitation and that even urban environments possess functioning ecosystems. With these boundaries dissolved, the primary model of landscape architecture becomes the ubiquitous pluralism of the city, rather than the isolated puritanism of the garden.<sup>119</sup>

At the same time there was also a shift in the ecological understanding of how ecological systems operate. When McHarg introduced his methods, ecological systems were generally held to operate under the ‘equilibrium paradigm’ in which natural systems are closed and self-regulating.<sup>120</sup> Under this ‘Clemensian’ view, a system exists as either a stable balance of organism populations and energy exchanges, or has been disturbed from this balance and is going through phases of succession to return to its previous state. Research started in the 1980s began to overturn this view, recognising that natural systems were prone to undergo both dramatic and gradual shifts, have

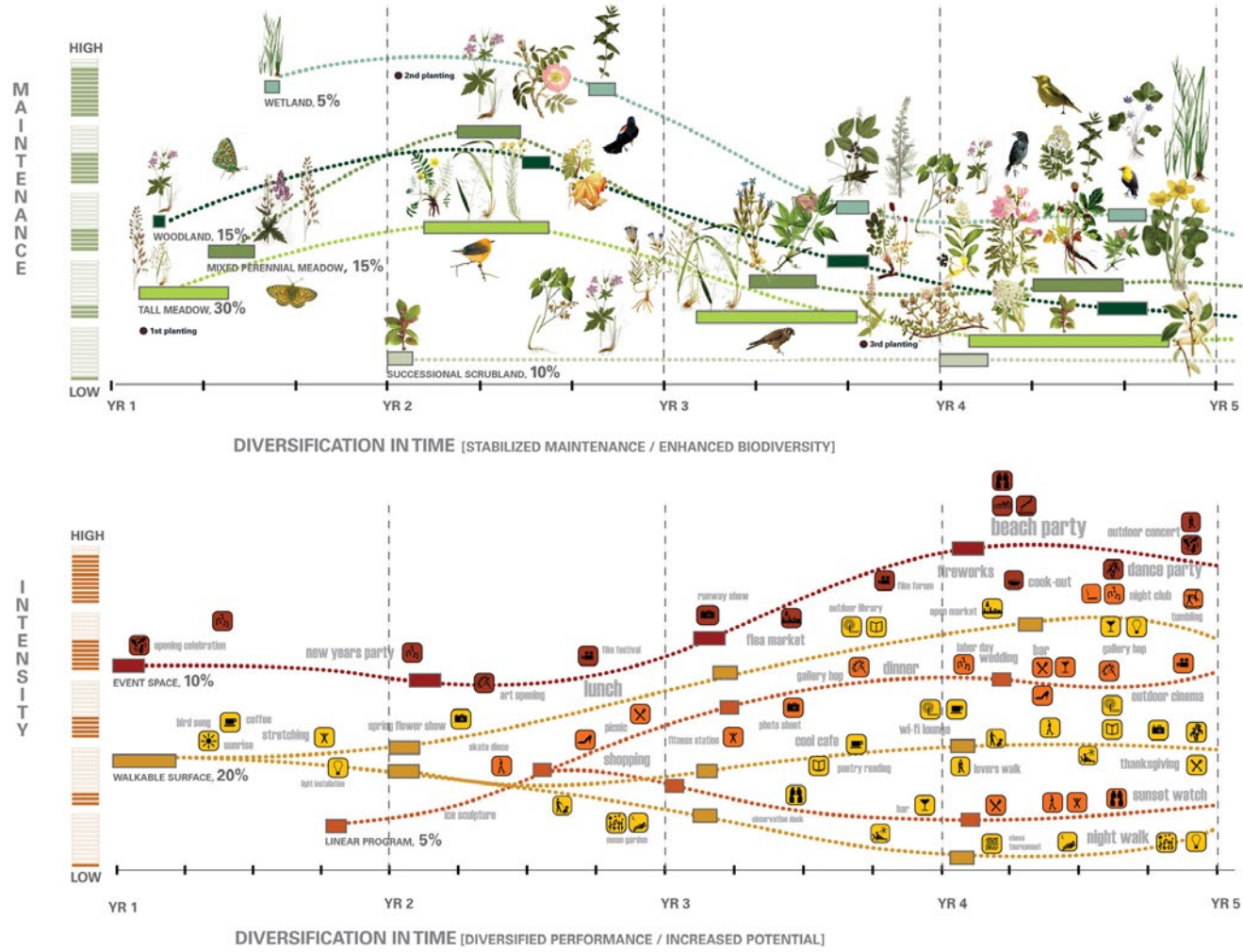


FIGURE 13. Diagrams detailing phased strategies for managing the ecological and programmatic aspects of the High Line.

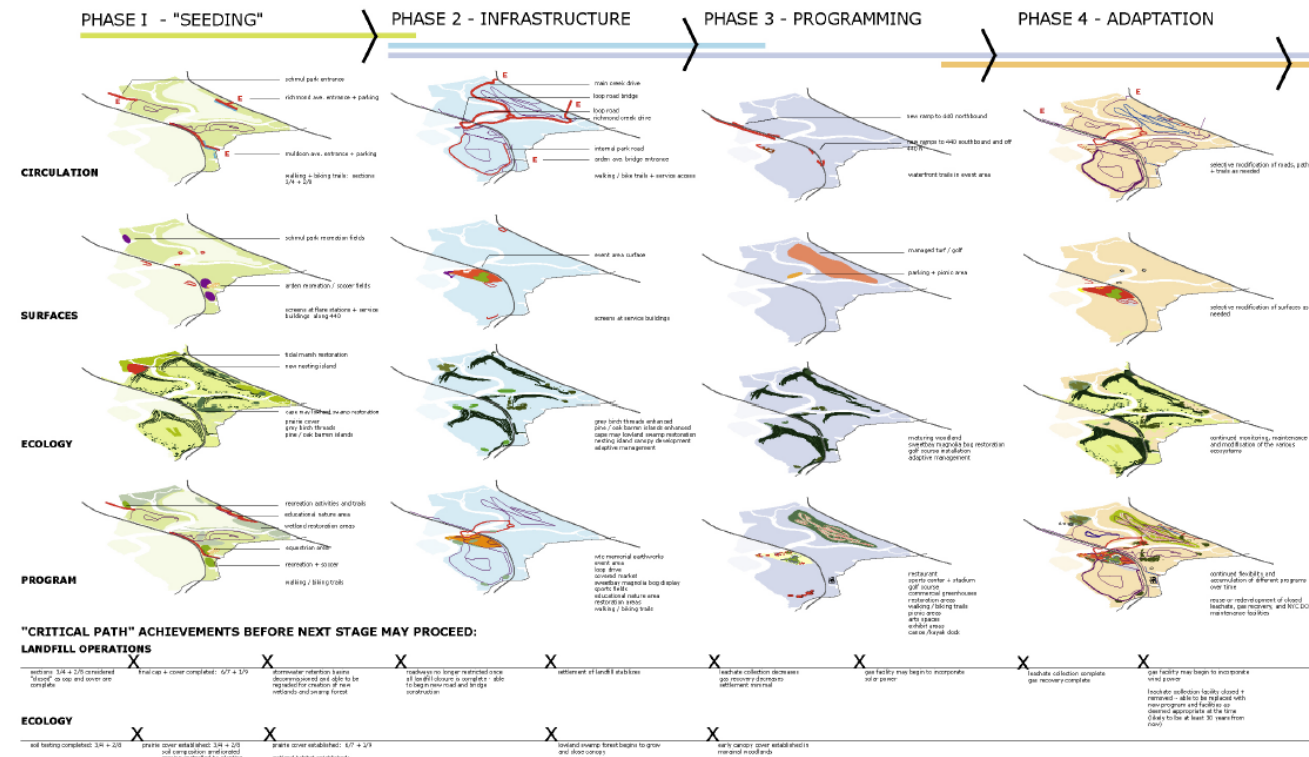


FIGURE 14. A board from the Jame Corner Field Operations entry for the Fresh Kills competition.

very imprecise boundaries, and are comprised of more complex interactions than previously thought. In this new paradigm, the role of disturbance and chance is key, as is a recognition that ecosystems do not have any one ‘true’ state that is inherent to their composition and context.<sup>121</sup>

Landscape architects followed this shift in ecologists’ thinking by creating designs that increasingly recognised the future evolution of a site. Landscape urbanist strategies give particular emphasis to this view of ecological systems as sites of complexity and chaos. Their designs give less emphasis to a particular ‘end state’ of a design, and instead favour a dynamic and flexible framework of possibilities that develops from a series of ‘seeding’ actions that take place throughout a project’s lifetime. These could take form as a phased planting strategy, or any number of other processual tactics, such as designing landform with an eye towards erosion, creating flexible event spaces, or using bioremediation.<sup>122</sup> As applied to urban parks, there is a shift from using maintenance plans to preserve a park’s initial appearance, and instead creating a design that accommodates change by working with the landscape’s existing systems.<sup>123</sup>

Similar techniques are used to fulfil programmatic requirements. Rather than use the architectural approach, where programmes are enabled through form, landscape urbanism looks to provide ‘soft’ infrastructures. These eschew fixed uses, and instead provide territories to be occupied by a number of expected and unexpected future uses.<sup>124</sup>

This emphasis on open landscape processes must often balance a contradiction between the possibilities of process-based strategies and the need to create forms that both enable and restrict possibilities. In the case of large-scale sites, such as Fresh Kills, the use of ‘seeding’ strategies relies on a slow changes eventually taking effect. However, in more heavily urban and small-scale sites these strategies can be less effective. Here there is a greater reliance on form and aesthetics to provide basic infrastructures and to define something more culturally and experientially engaging than a blank field.<sup>125,126,127</sup>

Even when process-based strategies seem promising, their assertiveness raises questions about how they will actually play out.<sup>128</sup> Without any mechanisms by which to test how a designed process will operate in reality, the future of a design that uses process-based strategies is vague and out of the designer’s control. To a certain extent, this uncertainty is a benefit, yet this openness needs to be balanced against a refined predictability that can test if, how, and why certain outcomes will occur.<sup>129</sup>

There are also problems in how the more landscape architectural landscape urbanists approach architectural form. While the role of the fundamental building block of urbanism may have changed, buildings cannot be ignored in favour of the landscape. In many urban plans, architecture is reduced to a set of statistics (waste water loads, shadows) and rendered as ghostly cubes sitting squarely upon a lush landscape.<sup>130</sup> The criticisms of architecture’s over-inflated role in defining urbanisms are certainly valid, but resorting to *lorum ipsum* buildings neuters the role of architecture. Some of these are surely just placeholder values for buildings that are yet to be designed, but the urban plan

nevertheless presumes and dictates the role of the building.<sup>131</sup> Here we have the opposite problem of the AA school’s approach: there is no exploration of architectural possibilities beyond the standard figure/ground forms that limit contextual engagement.<sup>132</sup>

### 3.6 The Root of the Problem

The criticisms of landscape urbanism elaborated above are symptomatic of a deeper issue that stems from a disjunct between rhetoric and representation.

Landscape urbanist theory dictates that landscapes are complex systems, and that our designs upon them should work in conjunction with the complex and processual nature of this medium. At present, landscape urbanism falls short of this ideal. This problem traces back to Corner’s claim that his maps deal with the virtual conditions present in the landscape. As noted earlier, Corner’s maps have an element of abstraction that obscures the reality of how a landscape operates. This is a flaw common to the wider practices of landscape urbanism and contemporary landscape architecture, where designers lack representations that help them understand and manipulate complex landscape phenomena in a comprehensive manner. Despite a claim that the field techniques of landscape urbanism are being imbued with specificity, the resulting analysis remains abstract.<sup>133</sup> Typically, this analysis then feeds into process-driven interventions which impose infrastructures and programmes that bear little relation to the pre-existing conditions of the landscape.<sup>134</sup>

While diagrams and phased drawings highlight the dynamic nature of such interventions, they all too often become self-referential aesthetic exotica which operate apart from a site.<sup>135,136</sup> The task of continuously analysing how the landscape itself interacts with the design is precluded when diagrammatic methods make only abstracted references to actual landscape characteristics.<sup>137</sup> Perversely, such landscape urbanist representations are productive because the imposed distance between design and site reduces the need to rigorously detail how the workings of the design will interface with the workings of the landscape.<sup>138</sup> Processual strategies become a way of avoiding complexity, rather than harnessing it.

This disconnect is also evident in the concepts of the ‘surface’ developed by landscape urbanists. These are all too often another excuse for abstraction. As a ‘surface’ the landscape becomes a plane to be operated upon, the better for it to serve as fertile ground for programmes imposed without reference to existing conditions or the inevitable effects of such events.<sup>139</sup> The site then becomes a means to an instrumental ends, an “abstract, homogeneous and continuous space.”<sup>140</sup> The freedom of an artificially-imposed horizontal homogeneity is nothing but a freedom to neuter the site and its workings in order to enact domesticating programmes upon the landscape. This tendency to create a blank canvas by way of tabula rasa<sup>141</sup> shows that there is yet more learning from La Villette to be done.

### 3.7 Towards a Landscape Architecture

There must be better forms of representation that encourage a connection between the new conditions imposed by a design, and the pre-existing conditions that are given by the site. Or, better yet, a flattening of the ‘new’ and the ‘existing’ into a single system that emphasises the role of design in actualising present potentials, rather than in imposing new novelties.<sup>142,143</sup> Despite the long history of landscape architecture and ecological principles, this form of synthesis within the design process remains limited.<sup>144</sup>

Conventional techniques, whether analogue or digital, are poorly equipped to represent the manifestation and interaction of landscape processes over time.<sup>145,146</sup> That the suggested solution to this quandary lies within the computer would seem counter-intuitive. As noted in the previous chapter, and will be further discussed later, the digital display is often a violent form of instrumentality. CAD software reduces the landscape to a 3D surface that displays a soulless uniformity, while Photoshop enables a surfeit of seductively pastoral propositions.<sup>147</sup> Rather than look to these established methods of digital design, landscape architects should look to computational design as a medium that is uniquely placed to drive an engagement with the complexities of the landscape.

It stands that the landscape architectural design process should frame and represent sites in the same terms as the sites themselves operate. This is what computational tools offer. At an epistemic level, such tools are uniquely capable of studying complex and dynamic processes<sup>148</sup> as the understanding required to construct computational rules directly maps to an understanding of the system itself. Here the virtual of Deleuze and the virtual of the machine collide because both operate through generative rules. Deleuze’s reality is an actualisation of *immanent* multiplicities while the computational is an actualisation of *designed* algorithms. Understanding this opens up a window into a reality defined by virtual systems; the very reality that landscape architects must inhabit when designing dynamic landscapes.

If anything, the affinity between these two virtual types is stronger when applied to the design of landscapes, rather than the design of buildings. Issues of scale and temporality are truly woven into landscape architecture.<sup>149</sup> Instead of the occasional focus upon diurnal cycles and mechanised elements, landscape architects must deal with scales that range from the garden to the ecosystem, along with fluxes that range from hours to decades.<sup>150</sup>

The dynamism and complexity of these phenomena meet their match in the dynamism and complexity of computation. The virtual of the computer imposes the flat ontology of Deleuze, creating an algorithmically driven plane of immanence where interrelationships are a matter of crafting the appropriate connections. Creating lines of flight through lines of code enables a true connectivity within the design process, rather than the ungrounded metaphors that are exploited to justify diagrammatic strategies. Here the relationships between data and diagram are real, each directly bringing the other into being, and allowing the designer to exploring a range of past, present, and future possibilities.

This hyper-connectivity comes from several features inherent to computational techniques.

Firstly, computational tools marry diagrammatic methods with pictorial representations, as discussed in earlier. For landscape urbanists, this offers an escape from the paradox between the technical rigour and creative abstraction when designing processual strategies. By translating the diagram into algorithmic rules, it is able create *real* relations between the existing site and the designer’s speculations upon it. Computation is unique here in its ability to resolve these speculations conceptually — by expressing the diagram’s generative rules — *and* technically — by precisely resolving these rules across multiple scenarios.

Secondly, computational tools are uniquely capable of managing multi-scalar relations. Existing landscape urbanist strategies are often characterised by disorienting leaps from the synoptic overview of the big plan down to the level of individual perception and materiality, missing a productive middle scale where the relationships between people and place occur.<sup>151</sup> When working in a digital environment, scale is an arbitrary and easily-changed factor. Computational tools allow for generative rules to be applied to an entire site, resolving details at all levels of scale. This enables an unprecedented level of interaction between general rules, and their evaluation against context-specific conditions.<sup>152</sup>

Thirdly, computational tools offer a unique capability to respond to and influence the complexity of a landscape. Using a digital model of the site, a simulation can be run that transforms the site in response to a design intervention. This newly-affected site can then be iterated upon through further simulations, or the original simulation altered and re-run against the original site conditions. The designer is empowered to think transversally across the available data, and to design within a feedback system between form, function, and time.<sup>153</sup> This actualises the shift from “object appearances to processes of formation” called for by Corner.<sup>154</sup>

This capacity to easily project a design through multiple states causes the implicit history of the design process to become explicit,<sup>155</sup> challenging the conventional “linear causality of design thinking”.<sup>156</sup> This is especially important when working with landscape systems, which typically exhibit non-linear qualities that require the designer to account for uncertainties in the existing landscape and the new design. The extreme elasticity of computational tools means that the ranges of possibility present within a landscape can be rapidly and rigorously resolved.<sup>157,158</sup> This is most evident when using computational techniques to examine the temporal aspects of a landscape. Here, time becomes just another parameter, a variable that is as easily edited as any other. The result is an unparalleled ability to evaluate design decisions across a range of time periods. Conventional tools can achieve similar results, but only through laborious reworking. This distinction between conventional and computation tools sounds minor, but the difference in ease is so extreme as to enable a design process that is qualitatively different, in which intuition and rigour can act in tandem.<sup>159</sup>



This is not to suggest that computation is the ultimate solution to all the design problems of landscape architects. Instead, it has great potential within an important niche: analysing and designing the operations of landscape systems. This undertaking is one that deals primarily with instrumental operations towards positivist ends. However, a computational approach examines landscape systems in terms of their underlying rules rather than their actual conditions. By contrasting these two aspects, the problematic determinism of McHargian methods is exposed as yet another abstraction.

It should be emphasised that the objective nature of these tools must work in conjunction with the subjective assessment of the designer. Computation offers the opportunity to evaluate and interlink performance criteria, but aesthetic effects must be evaluated using the designer's judgement. Tools are not without consequences, and in many respects 3D-modelling is not the best environment to generate and assess affective qualities, especially when designing landscapes. Designers must be wary of these limitations, and address them by looking towards hybrid practices that integrate both new and conventional representational techniques. Similarly, while the rules of computation are rigorous unto themselves, their results rely on the correct rules being chosen. There is a danger in assuming that a simulation offers a perfect crystal ball, rather than a series of probabilities. In most use cases, good simulations will create results more rigorous than traditional techniques. However, opportunities should be seized to work collaboratively with specialists that can assess and improve the results of computational techniques.

### 3.8 Conclusion

This chapter has introduced the context for the contemporary emergence of 'landscape urbanism' as a congruence of changing attitudes to urban planning, landscape ecology, landscape representation, and architectural design. It identifies two strains of thought within these practices. The first is characterised by an approach that designs landscapes by way of computationally designed architectonic forms. The second is characterised by an approach that designs landscapes by way of processual diagrams that enable ecological and programmatic changes.

Common criticisms of both modes have been raised, as well as a deeper problem with representational methods that create alienating abstractions. Computation has been posed as a remedy in that it enables new forms of representation that better deal with complexity and thus reduce the need for such abstractions. Landscape urbanism is already attentive to the role of representation, and well versed in processual thinking. It is fertile ground in which to pursue a computational landscape architecture.

4 *Another Pattern Language*

In the previous chapter I suggested that computational thinking and computational tools could construct new forms of representation that better engage with landscapes.

This chapter introduces a methodology for a series of experiments that test the intersection of computational design and landscape architecture. It first introduces a ‘pattern’ framework that addresses common issues with how computational techniques are used and propagated. I then discuss the patterns I’ve developed to test this methodology. Each of my patterns are described in terms of how they operate, and what they offer landscape architects.

4.1 **Patterns Past and Present**

The notion of design patterns has had a varied history across a variety of fields. Originally coined by Christopher Alexander, they were later adopted by software engineers, web designers, and architects working with computational tools. This history is briefly explained in order to distinguish the salient features of my patterns developed from those of other implementations.

Programming’s chief challenge is complexity. Its solves complexity through abstraction. Given computational modeling’s reliance on programming, the challenges of complexity and abstraction have entered into the design process.<sup>160</sup> To help address this conflict Robert Woodbury has published *Elements of Parametric Design*. Woodbury examines common problems within computational design, and suggests a series of patterns that act as best-practices for ameliorating technical and conceptual complexities. In this way each of his patterns act as a “generic solution to a shared problem.”<sup>161</sup> For example his ‘jig’ pattern describes a process for building “simple abstract frameworks to isolate structure and location from geometric detail.”<sup>162</sup> Following this pattern a designer could create a morphed tube shape by using a few key curves to define a skeletal outline that is then covered with an enveloping skin. In this way a series of simple and easily-modifiable curves can be used to construct a more complex volume.

Woodbury’s patterns largely build upon the concept of design patterns that was developed in software engineering.<sup>163</sup> Within this discipline, design patterns usually help address issues of performance and speed, or issues of complexity that emerge from writing and understanding code. For example, a programmer would use the ‘proxy’ pattern by creating an object that represents certain characteristics of another object, such as an image placed within a document.<sup>164</sup> Displaying a complex image is potentially a computationally taxing task, so the use of a proxy enables a document to be displayed faster because the proxy represent an image’s position in the page’s layout, but delays its loading until after the document is initially opened.<sup>165</sup>

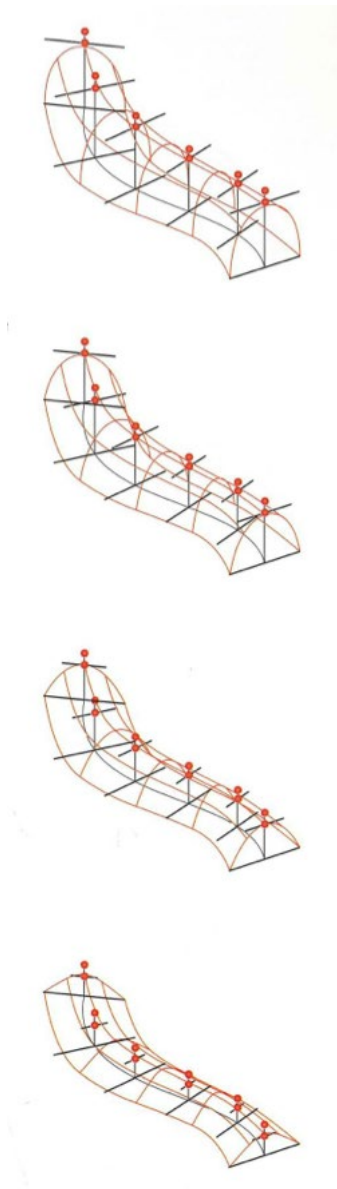


FIGURE 15. Woodbury’s diagram documenting the jig pattern.

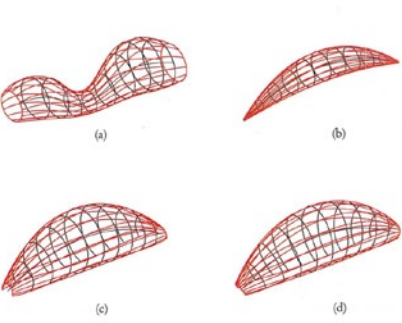


FIGURE 16. An example of a jig in action.

The patterns in *Elements of Parametric Design* act in a similar manner, but directly address the unique problems that emerge when programming within 3D modeling software. Here issues of performance and code quality are intermingled with the mathematics used to define geometry. Despite the focus on geometry, it is important to note that each of Woodbury’s patterns does not create a specific form. Instead they are a set of higher level rules that guide the lower level rules that then create particular shapes. Woodbury’s approach tells the designer how they should best approach common problems that arise when programming, not what shape their design should take.

In contrast, Christopher Alexander’s patterns aim to solve the very concrete problems of our built environment. They emerge from his pioneering work during the 1960s, where attempts to use computers to solve architectural design problems led him to approach all design problems using the logic and structure of computing.<sup>166,167</sup> From this perspective he came to see the built environment as a complex system of recurring and interlinked problems. Alexander’s best-selling<sup>168</sup> *A Pattern Language* introduced design patterns as a methodology for breaking down each of these problems and detailing their solutions. Each pattern describes a recurring problem and a resulting solution such as “Sleeping to the East” in regards to bedroom layouts, or creating an “Entrance Transition” when designing a front door.<sup>169,170</sup> When applied collectively, these patterns form a comprehensive guide to constructing towns and cities.<sup>171</sup> The messy complexities of a system (the built environment) could be abstracted away by creating a much simpler system (the patterns) of logical rules.

While many of Alexander’s patterns represents keen insights into best practices and an eye for effective details, many represent quite significant limitations on the scope of the designer, the planner, and the community, such as:

- Buildings must be less than four storeys;
- Interiors must be painted warm colours;
- Comfortable spaces require thin columns;
- Every neighbourhood should have at least one corner grocery.

The level of formal prescriptivism in these patterns and others undermines Alexander’s attempt to establish a series of generative rules for planning the built environment, and lapses into a supposedly objective set of prescriptions for designing a retrograded utopia of the vernacular.<sup>172,173</sup> While their goals are laudable, and many of the patterns credible, creating strict formal definitions ignores the possibility that new typologies may be effective and that new typologies usually exist because they are needed.<sup>174</sup> There can be no ‘timeless way of building’ unless we only encounter timeless problems.

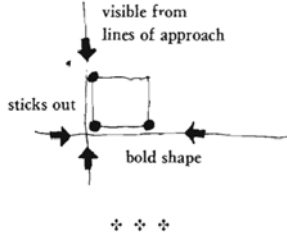


FIGURE 17. Diagram for the Entrance Transition pattern.

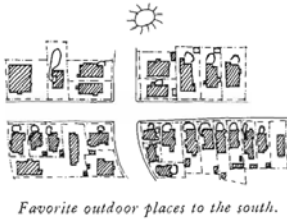


FIGURE 18. Diagram for the South Facing pattern.

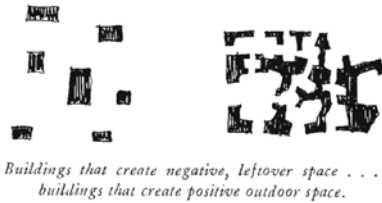


FIGURE 19. Diagram for the Positive Outdoor Space pattern.

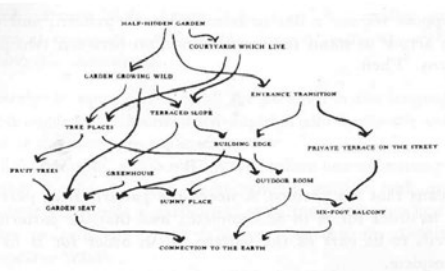


FIGURE 20. It’s patterns all the way down: diagram showing the interconnected nature of the patterns.

## 4.2 Landscape Patterns

My patterns draw something from Woodbury’s patterns, from Christopher Alexander’s patterns, and from open-source code libraries. Like software design patterns, my patterns do not propose particular solutions to particular problems; instead they are tools that guide the development of solutions. Like Alexander, my patterns work within a disciplinary context in that they address problems that are landscape architectural, such as issues of hydrology and ecology. Unlike Alexander, my patterns do not prescribe form.

Instead my patterns aid the user in analysing and simulating landscape phenomena within the design process. In this way they operate in a similar fashion to Woodbury’s patterns, except instead of dealing with problems that arise within computational design, they focus upon the problems that are specific to a design discipline. As an example, two of my patterns look at issues related to hydrology. One pattern enables the user to analyse the flow of water across a surface, while the other presents the effects of tidal cycles and sea level change across a site. These could be used as part of a site analysis or to assess the capabilities of a design.

Because my patterns can be used in tandem with each other, and in conjunction with standard computational techniques, they can be used as generative tools that integrate analysis, simulation, and design intent into a feedback loop. For example, the pattern that analyses the flow of water could be linked to a piece of code that generates landform, which then enables the designer to rapidly create a grading profile that ensures swales capture storm water. The pattern that analyses tidal levels could be linked to a piece of code that defines a seawall with embedded rock-pools to check that each pool gains the requisite tidal flushing. In this way my patterns can augment and accelerate existing design practices, as well as providing a framework for creating new computationally-driven design techniques in landscape architecture.

Technically, my patterns implement a ‘modular’ structure. In programming, modules are collections of code that perform certain functions, often available online as open-source software.<sup>175</sup> If a programmer needs to achieve a certain task, they could integrate a module into their program rather than recreate the desired functionality from scratch. Modules typically make it easy for the programmer to ignore how the module’s code actually works, and instead ‘ask’ it to perform simple tasks and return the result. One particular module might handle the task of compressing a file into a ‘zipped’ form to reduce its size. Rather than recreate the algorithms used to do this, a programmer might download a module that performs this function, include it within his project, and then tell his code to call upon this module whenever it needs to compress a file in order to better send it over email.<sup>176</sup>

By adopting a modular structure, my patterns aim to integrate easily into the design process regardless of the user’s programming skill. Novice users of computational modeling tools understand the basics of working with geometries, such as how to create a series of circles with varying widths. What they tend to struggle with is more complicated models, where complex

geometries (such as surface parameters, or vectors) are interlinked and the interrelationships become hard to discern. To help ameliorate this complexity, the modular nature of my patterns means they output a small amount of clear information, typically in the form of numbers or simple geometries. More advanced users will be able to modify the code of each module itself to exert full control over the desired functionality.

The use of a modular structure also assists in creating more comprehensible computational models. By breaking down a model into parts that can each be resolved individually and linked back into a wide schema, the designer can resolve each piece in detail without inadvertently effecting other parts of the model.<sup>177</sup> Computational modeling — especially using parametric techniques — requires this ‘divide and conquer’ strategy if the overall composition is to remain comprehensible.<sup>178</sup>

In this way my pattern approach enables a ‘loose coupling’ between design elements, whereby each element is minimally dependent on the others. This reduces ripple effects where small changes in one element create unexpected and undesirable effects on other elements. By ensuring that each pattern is as self-contained as possible, and the connections to other elements are deliberate and minimal, the scope for errors is reduced. This follows the idea expressed in a common software engineering aphorism: “be conservative in what you send, liberal in what you accept.” In this case that means the inputs into a pattern (pieces of geometry and/or particular variables) should encompass a wide range of possible values, while the output data should be consistent and recognisable.<sup>179</sup>

As demonstrated in software engineering, this modular approach has great potential in reducing the challenges of complexity. Many designers lack experience in traditional programming, so increasing the readability and simplicity of code makes computational design techniques more easily understood and applied. In a world where code is increasingly shared, copied, pasted, and applied,<sup>180</sup> the tendency to reuse rather than rebuild means that having access to quality code is a key community resource that benefits novice and advanced users alike.<sup>181</sup>

## 4.3 Programmed Patterns

My patterns are implemented using Grasshopper, a popular plugin available for the Rhinoceros3D (Rhino) program. This particular setup was chosen for a number of reasons:

- The combination of Rhino and Grasshopper is well established within architectural practice and education.
- Novices and experts both use Grasshopper because its graphic, rather than text-based, interface provides efficiency and accessibility. Rather than creating computer code using a textual scripting language, users create a mind-map like diagram by selecting components and drawing connections between them. These schema are then automatically and invisibly converted into traditional text-based code so that the 3D modeling program can create the required geometry.



- Because Grasshopper operates within a powerful 3D modelling program, users can mix and match between typical CAD tools and computational techniques.

However the program has several limitations:

- While the graphical interface is great for novice users, and for quickly creating simple models, there are several trade-offs in terms of understandability in comparison to a text-based method of programming. The flowchart-like schematics constructed in Grasshopper are poor abstractions of software structure because data does not always flow according to the linear hierarchy that Grasshopper enforces. In most programs, multiple pieces of data pass through different areas of code at different times in different orders depending on contextual conditions. Attempting to emulate this in Grasshopper creates schema that are hard to comprehend given the complex topologies required and the large amounts of redundancy that inevitably occur.<sup>182</sup>
- Similarly, the graphical interface does not support many advanced programming features, mostly in an effort to keep the ‘data flow’ metaphor comprehensible.<sup>183</sup> Reasonably standard features such as modules, loops and object-oriented methods are not available. Utilising these, and other, features can further reduce the complexity of Grasshopper schema.
- Similar to the use of predefined tools in CAD programs, Grasshopper definitions are created by combining prefigured components. This limitation is much less severe than in traditional CAD software, as Grasshopper components are less prescriptive and can be modified to a much higher degree. However there are still particular elements that the popularity of Grasshopper has made into clichés, such as the voronoi pattern.

While my patterns are opened and edited in Grasshopper, they are created using a hybrid approach that combines the graphic elements of standard Grasshopper definitions with pieces of textual code embedded in ‘custom components.’ This approach is possible using Visual Basic and C++ scripts, but I have used scripts written in the Python<sup>184</sup> language that was made available in the most recent release of Rhinoceros. This hybrid approach has a number of benefits:

- The ability to make use of advanced programming features[ available in textual languages that accelerate development and provide new capabilities.<sup>185</sup>
- The ability to ‘hide’ complex computational operations in custom components which dramatically cuts down on the complexity of the Grasshopper definition. Novice users can ignore these pieces safely, while they are still present and modifiable for those that wish to do so.
- Increased portability because blocks of Python functionality are more easily separated and reused.

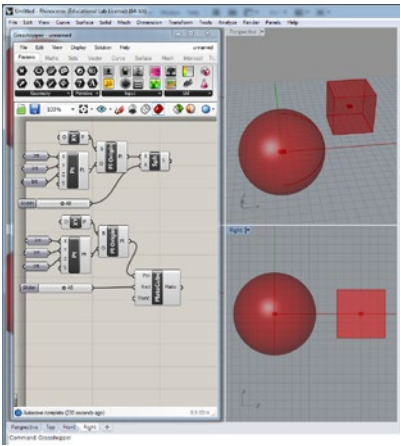


FIGURE 21. A Grasshopper definition to create a sphere and a cube, as displayed in Rhino.

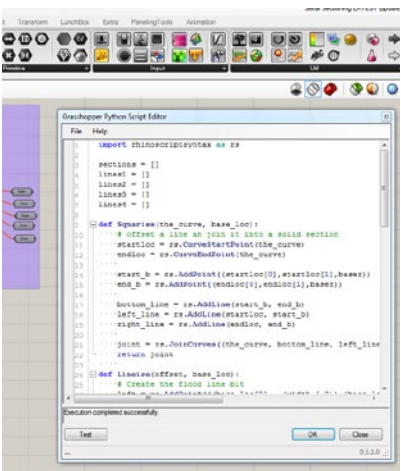


FIGURE 22. A Python script contained inside a Python component, which is in turn created within Grasshopper, a plugin that operates inside Rhino.

#### 4.4 The Patterns

Each of the patterns produced over the course of my research will now be described. The first two deal with issues of representation in landscape architecture, followed by a pattern that focuses upon ecological conditions in relation to plants. The final two patterns are used to investigate hydrological phenomena.

#### 4.5 Fields

This pattern aims to improve the display of information within a 3D modeling program. It achieves two particular functions.

This first is to display quantitative information within an environment that privileges 3D forms. It implements some of the visualisation functionality common in GIS systems, but allows it to operate within a fluid 3D environment rather than a comparatively clumsy 2D program. This analysis then becomes more useful as it can be used at any stage of the design process.

The chosen method of displaying quantitative information attempts a style of visualisation that improves upon the synoptic ‘layer cake’ methods prevalent in GIS systems. Rather than polygonal shapes, my pattern generates a grid populated by a number of circles, whereby the size and colour of the circles represent the information present at that spatial location.

The hue of each circle represents the type of information represented, such as *residential building* or *mudflats*. This allows for polychromatic circles that describe liminal areas of transition between conditions, such as the transition between shallow water, mudflats, and bank. This blending can also accurately depict areas of overlap, such as a displaying a mixed-used building as a purple combination of commercial (red) and residential (blue) programs.

The colour saturation and size of the circle represents the quantity of the information present, such as *native plants* or *slope angle*. This allows for the display of graduated effects, or intensive conditions, where there are shifts between particular quantities. For example, when measuring the population of native plants, a spectrum from dense to sparse can be visualised, rather than displaying a single region which would suggest a spatial dichotomy.

This pattern is operated via several steps:

- The user select pieces of 2D geometry that belong to a particular typology. To display quantitative intensity within a typology, the pieces of geometry must overlap or stack. So a circular area which has a large quantity of native plants at its centre, and few at its periphery, would be defined by series of overlaid circles that successively shrink.
- This selection process is repeated for each typology.
- The user defines a boundary that delimits the grid.
- The user then selects the degree of detail required by adjusting the grid’s spacing parameter.

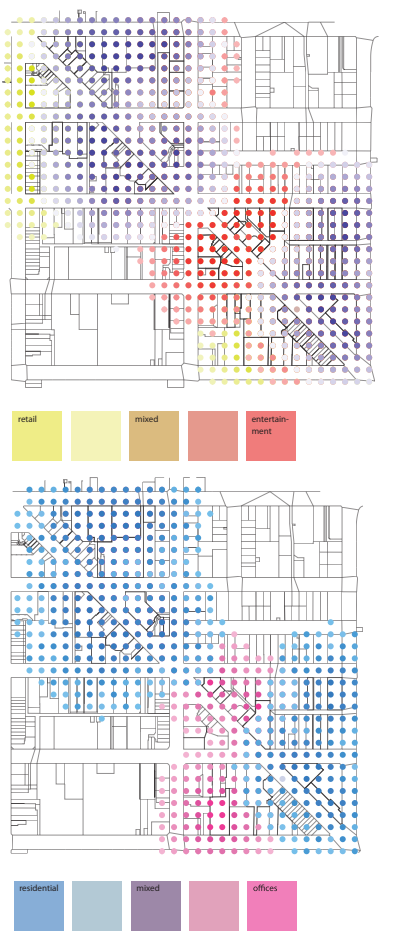


FIGURE 23. An early version of the Fields pattern, applied to the display of building programmes.



- Results are then generated and displayed. Moving pieces of geometry will then automatically update the results.

This visualisation technique attempts to combine the clarity of the synoptic map with the detail of a medium-scale study. Viewed at a large scale, this type of ‘field map’ grants an overview across a number of data sets because the size, saturation, and colour of the circles form global patterns that are easily understood in aggregate. When viewed at a smaller scale, the field reduces to a more precise set of local relationships that allow the relationships within a spatial location to be understood.

Like any form of mapping, this visualisation technique is ultimately only as good as its data. However, even when using GIS-style polygons as its data source, it re-represents them in a manner that depicts landscape phenomena as gradiented conditions, not segmented shapes.

#### 4.6 Flows

This pattern simulates the flow of water across a surface. Such flows are typically understood intuitively when working with simple, contoured forms at a large scale. However, when working with a 3D modeling program, landforms are typically represented using mesh or surface geometries that make changes in grade more difficult to comprehend, particularly at smaller scales.

This script precisely simulates the flow of water across a surface by creating a series of points that represent a landing point of a ‘drop’ of water, which then travels downhill to create lines that represent the flow path of the drop. This enables the designer to explicitly, rather than implicitly, understand how the topography of the land affects water drainage and related issues.

The pattern is operated through several steps:

- A piece of 3D geometry is selected as the terrain.
- The script evenly distributes points on the surface in a grid pattern, each representing the landing point of a drop of water. The distance between the points can be modified according to the scale and level of precision needed.
- The script extrapolates where these landed drops would flow to based on an analysis of the terrain. A composite line represents the flow path.
- These lines continue until the ‘drops’ reach a basin (a well in the surface they cannot escape) or until a user-specified cut-off point (for example 50 meters).

Technically, the script uses an agent-based simulation methodology. Each ‘drop’ follows the same logic whereby its position is evaluated and it proceeds to follow the downward curvature of the surface. This logic applies iteratively for each individual drop until a final resting point is found, or a cut-off limit is reached. This agent-based approach means that, acting in combination, the system generates all possible behaviours (water flows) within a delimited ‘search space.’<sup>186</sup>

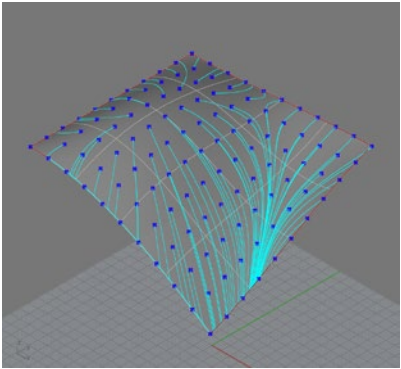


FIGURE 24. The Flows pattern, showing the movement of water across a surface.

That said, this is a relatively imprecise method. It assumes that the flow of water take place in an idealised environment where the surface is impermeable and that each drop has the momentum needed to travel down even a slight gradient. For this reason it is not an accurate method for precisely simulating the quantity of water flow. But it serves a useful role in that these concerns — even if in an imprecise form — can be made much more active throughout the design process. Even an approximate understanding of how water would drain helps suggest landform and material changes throughout the design process, rather than having to re-factor a finalised design to deal with unconsidered drainage issues.

#### 4.7 Levels

This pattern simulates water levels over time, focusing on the short-term and long-term changes that come from tidal cycles, sea level rise, and flooding. These changes are usually modeled using geometric planes that can be raised and lowered to simulate a body of water. While this approach is effective in displaying simple changes, it is imprecise and limiting because it cannot easily switch between different kinds of level changes that occur over different time periods. This pattern takes this typical approach and augments it by setting up a simple workflow for simulating multiple types of changes acting across diverse time periods. It operates through several steps:

- The user selects a piece of geometry that represents a body of water, such as an ocean, river or lake.
- Multiple parameters allow the user to input the expected type and quantity of water rise. There three types available cover tidal cycles, sea level rise, and flooding.
- Sliders are available that represent hours, days, months and years. Flooding is typically simulated over the course of hours; tides using hours, days and months; and sea level rise over the course of years.
- A contour is projected on the 3D geometry that represents the new water level according to the time and type sliders. Updating the geometry or any of the sliders recalculates the water level.

As with the previous pattern, this process serves as a somewhat accurate and easily utilised method for evaluating a design against changes in water level. Its ease of use means that the heights and shapes of design features can be assessed against tides, flooding, and sea level rise earlier in the design process, enabling flaws and opportunities to be discovered earlier.

#### 4.8 Sections

3D modeling programs are typically operated in perspective, with the option to lock a viewport into a plan, elevation, or isometric view. Sectional view are possible, but often laborious to set up because section planes must usually be positioned as if they were pieces of geometry.

This pattern allows the user to generate a number of sections through a particular area. The sections are then displayed apart from the 3D geometry so that they can be examined in tandem.

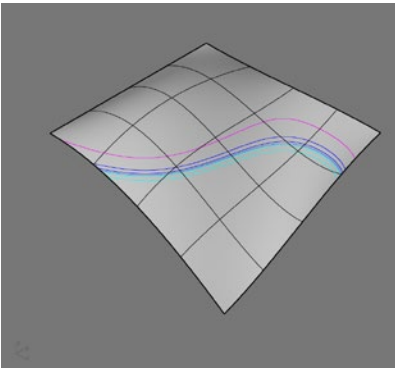


FIGURE 26. The Levels pattern, showing multiple types of water levels, and how they would intersect with an existing terrain.

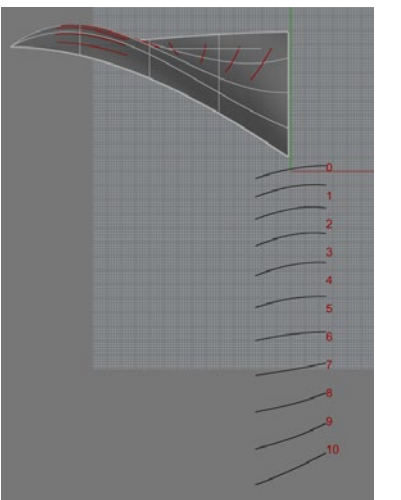


FIGURE 25. The Sections pattern, showing section cuts through a surface, and the arranged sections.

The pattern is operated through several steps:

- The user places two lines which represent the bounds of the sections.
- A parameter is adjusted that determines the number of section lines to be evenly distributed between the bounds.
- The intersections between the lines and the geometry are calculated, and the sections are then spaced serially at the origin of the model space.
- If any of the geometry changes then the sections automatically update.

While this is a relatively trivial technique, the ability to easily create serial sections is often extremely useful. As a representational technique, sections effectively depict landform, particularly when operating in a digital modeling environment where geometry is depicted as obscuringly shaded shapes or confusingly tangled wireframes.

## 4.9 Seeds

This pattern focuses on the design of planting plans and schedules. It attempts to ameliorate a number of problems typical to this task:

- Representing, much less accurately depicting, plants within a 3D modeling environment is difficult. The use of either 2D placeholders (ie circles) or placeholder models (fully rendered, but not native or accurately sized species) is common. The even more common approach is to put off the issue until the design is mostly resolved.
- Creating planting tables and detailed planting plans is a laborious task. Again, it is usually left until the rest of the design is more resolved so that any design decisions do not need to be reworked.
- Representations that depict the growth processes of plants are largely absent from most representations and models, with plants typically depicted in their mature state.
- Designing the distribution and selection of plants is usually performed *after* the design is reasonably resolved and the contextual conditions of the planted areas are known.<sup>187</sup> For example the quantity of light, or grade of a slope, are resolved first, and the planting list developed as a response.

This pattern aims to solve these problems by setting up a simple workflow for creating and simulating planting growth. The pattern is operated through several steps:

- A species schedule is created in Excel. Each species is assigned a row and various attributes are specified, such as labels (common name, species name, indigenous name); growth characteristics (initial and mature values for a species height, width, root radius); growth variability (how much variety would be expected in the above growth characteristics); plant spacing; slope tolerances; soil saturation tolerances; cost; colour; and links to textures that depict the plant in plan and section at various stages of growth.
- The user selects particular pieces of geometry that have been created in Grasshopper or Rhino, such as a planting bed, a river bank, or a roof.

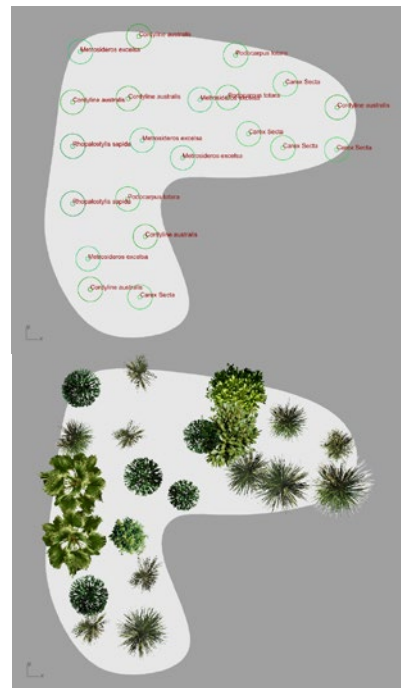


FIGURE 27. A simple planting plan, generated by the Seeds pattern, showing a technical plan view, and a render plan view using species-specific images.

- The script distributes species within the selected shapes. A parameter lets the user decide whether the distribution should follow a random distribution, a circle-packing pattern, a square grid, a hexagonal grid, or a triangular grid. The selection and spacing of each particular plant takes into account the characteristics specified in the Excel file. Because the selection of species and their locations is quasi-random, this process can be repeated to generate different results.
- Depending on another parameter value, the plants are either represented as 2D circles, 2D textures, 2D circles with a 'pipe' to roughly simulate their height, or as rendered 3D textures.<sup>188</sup>
- Initially the species are shown at their 'initial' stage of growth – a state that represents the typical width, height, and spacing characteristics of a species when it is ready to be transposed from a nursery and planted on site. A slider within Grasshopper represents the current year, ranging from 0 (the present) to 100 (a century ahead). Dragging the slider increases the width/height values of the plants according to their growth attributes
- The planting table, the planted area, and the time value can all be modified at any time to automatically update the results. Multiple copies of the pattern can be run in tandem to match multiple collections of species to multiple locations within a site.

As a result, this pattern enables the designer to create a planting plan according to a combination of objective and subjective factors in a manner that is much easier than when following traditional techniques.<sup>189</sup> By automating the laborious aspects of selecting, distributing and representing plants, fully-resolved plans can be introduced into the design process from the outset, and developed more extensively as the design process progresses. Additionally the pattern can be modified to take into account environmental analysis from the 3D model and other Grasshopper definitions, creating a feedback loop between landform changes, their ecological effects, and the vegetation that would best suit these conditions while satisfying aesthetic and/or performative criteria.

## 4.10 Conclusion

This chapter has introduced the primary output of this thesis: the patterns. My patterns aim to appropriate computational tools for landscape architectural ends, and thus introduce a new way of working that drives a more powerful engagement with natural systems.

Their aims accord with the goals established in the previous chapter. Each pattern looks at a dynamic landscape phenomena and makes its dynamism explicit by exposing its rules. The patterns also look at particular visualisation techniques in landscape architecture, such as the section and the map, and examine how these can engage more with the ‘middle ground’ of the landscape, particularly in the context of a 3D modeling environment that privileges the perspective view.

While the capabilities of the pattern may be modest compared to the range of tools offered in specialist software, their ability to be used within a common 3D-modelling package makes them much more useful to designers. Moreover they are designed to address many of the common problems that make computational models complex for both novices and experts alike. Further, by publishing the patterns online as open source software it is hoped that they will improve over time through successive improvement by myself and other collaborators.<sup>190</sup>

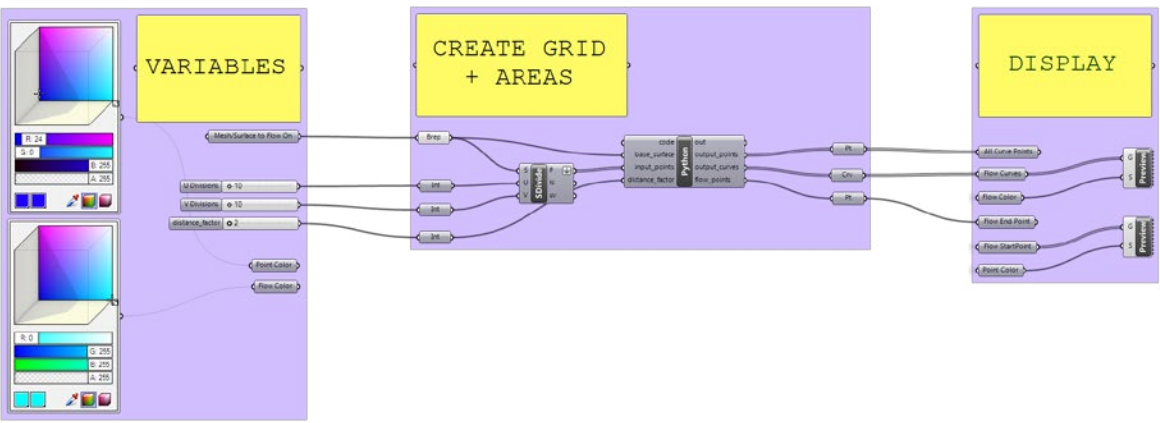


FIGURE 28. The Grasshopper definition for the Flows pattern.

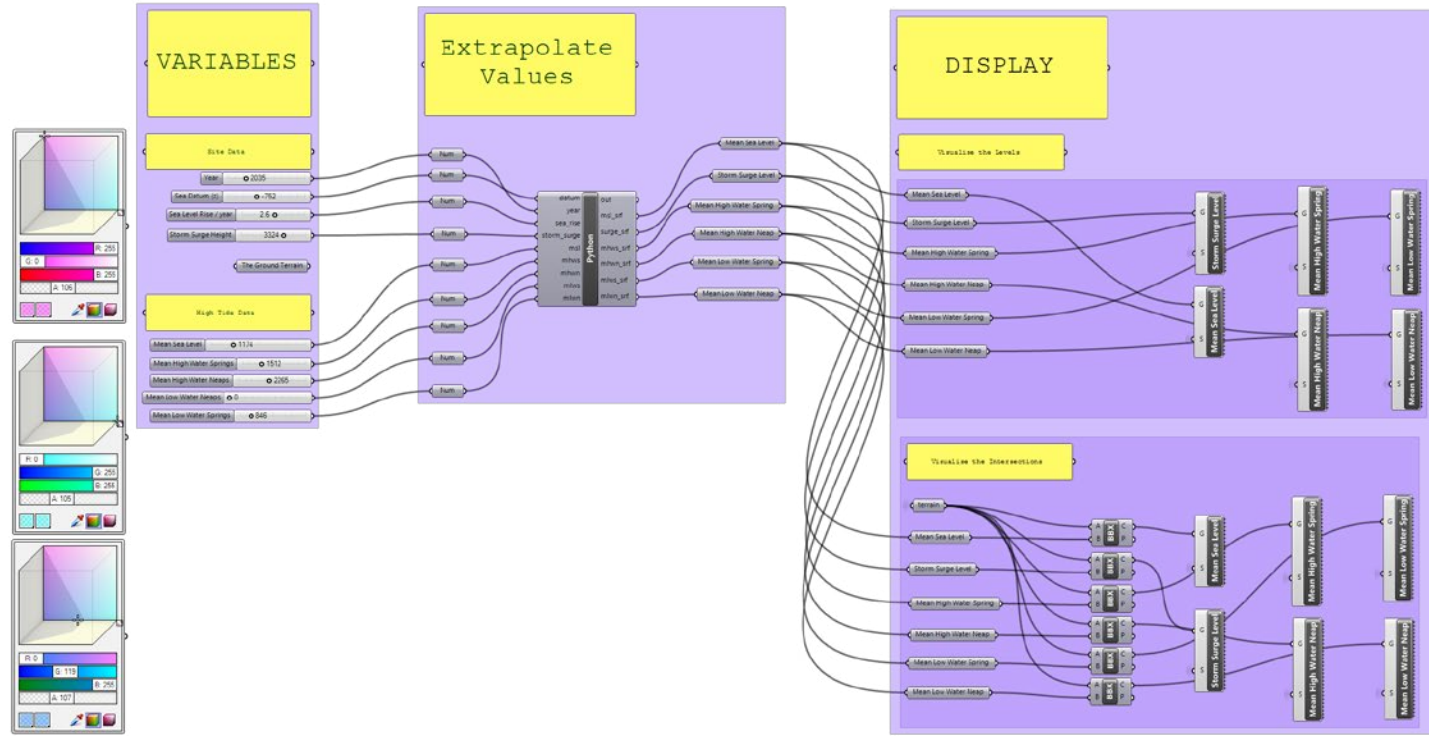


FIGURE 29. The Grasshopper definition for the Levels pattern.



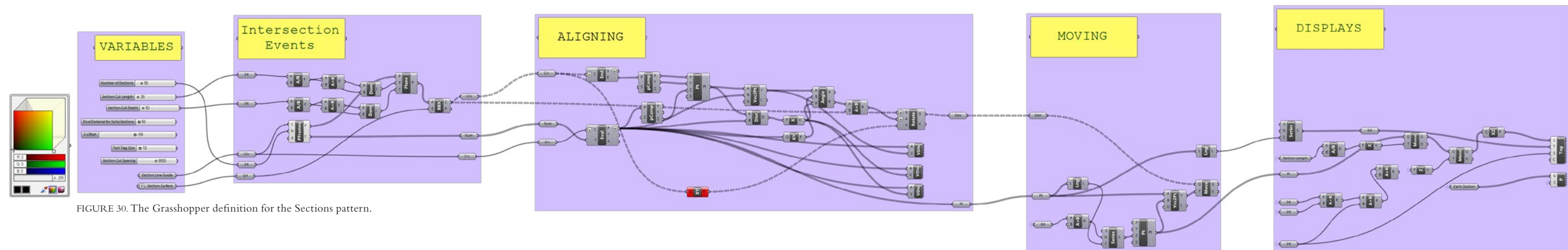


FIGURE 30. The Grasshopper definition for the Sections pattern.

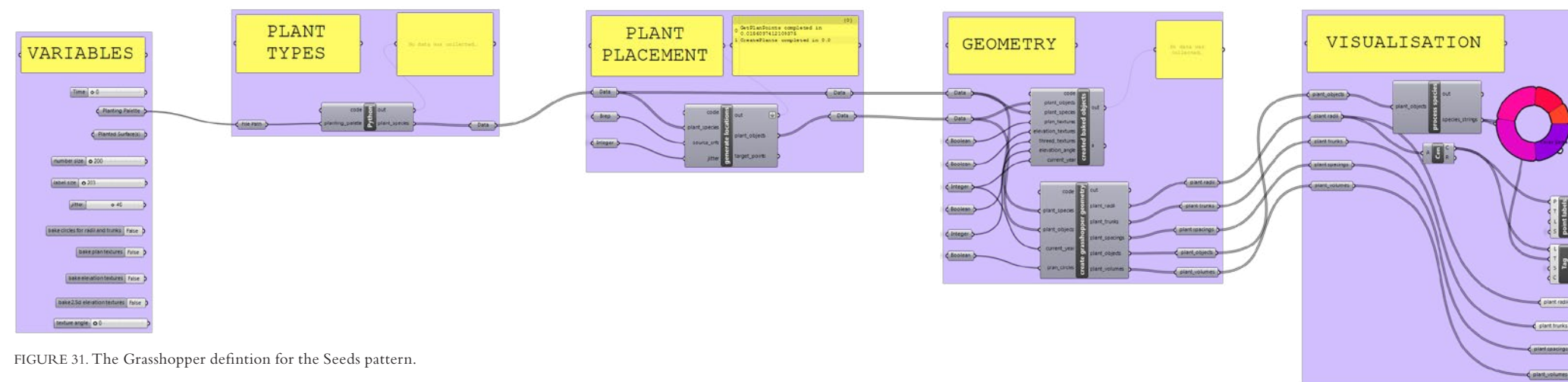


FIGURE 31. The Grasshopper definition for the Seeds pattern.

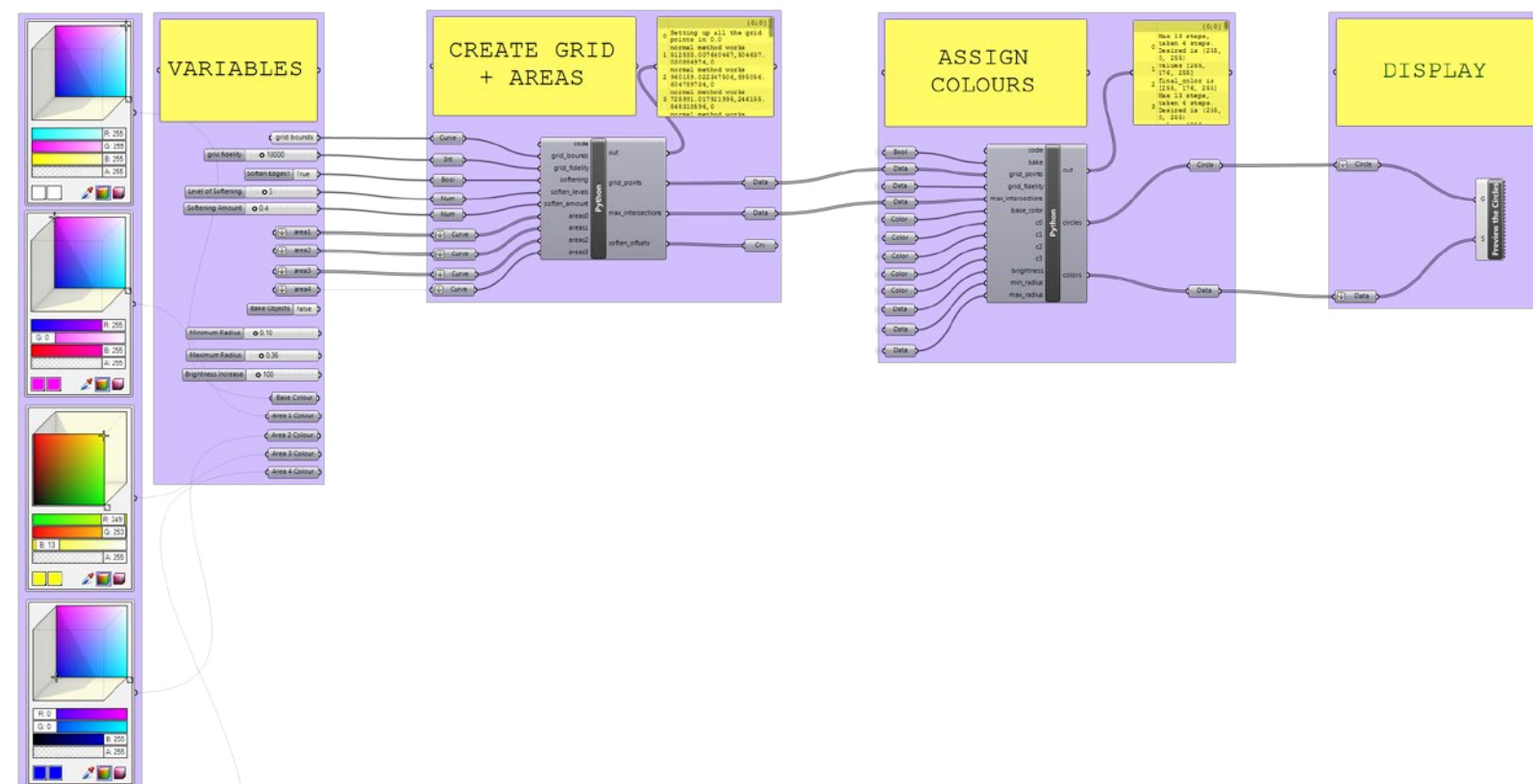


FIGURE 32. The Grasshopper definition for the Fields Pattern.



## 5 Case Study: Seaview

In this chapter I discuss how my patterns operate within the design process. Having selected a site and brief, I created an initial and revised concept design. Following landscape urbanist strategies, I use the patterns in conjunction with other computational techniques to create a design that leverages landscape systems. The utility and limitations of the patterns are then evaluated.

### 5.1 The Brief

Seaview is a suburb in Lower Hutt, Wellington, New Zealand (Figures 34 & 35). It has been a predominantly industrial suburb over the last century, and currently contains approximately half of the industrial floor space in the Wellington region. Petrochemical companies occupy the largest plots, while the remaining area includes manufacturers, research centres, artist's workshops, and a cattery, amongst other businesses. The waterfront and the Waiwhetu River are the primary landscape features, but both have been dramatically affected by industrial use.

My design begins with the goals set out in *Vision Seaview Gracefield 2030*, a plan developed in a collaboration between the local Council and local businesses. The key themes of the report are a desire to improve recreational opportunities, infrastructure, and the environment. Proposals mentioned in the report include:

- Developing long-term wetland areas adjacent to the Waiwhetu Stream, as well as a “cultural history and arts trail.”<sup>191</sup>
- Enhancing and building upon the area’s industrial character.<sup>192</sup>
- Numerous other projects aimed at improving environmental health and amenity.<sup>193</sup>

Across Seaview there are two significant environmental challenges. The first is erosion, a major threat that occurs on both on the waterfront and on the banks of the Waiwhetu. This is particularly harmful to the waterfront’s recreational value and to the stability of the adjacent road.<sup>194</sup>

The Waiwhetu River was channelled in 2012 as a means to reduce flood risk, the second major environmental challenge. The channelling has proved effective at reducing flooding, but has increased erosion and severely affected the riparian ecology: “the majority of the estuary now has steep intertidal margins which greatly limit the area where salt marsh is able to grow ... as a consequence of the narrowed flow channel, flow velocities are relatively high and the planted margins have been subject to erosion that has washed away many plants, undercut banks, and eroded sediments.”<sup>195</sup> Portions of the banks and the river surface were dredged and capped to remove industrial pollution, but the recent erosion has exposed some of the contaminants.<sup>196</sup>

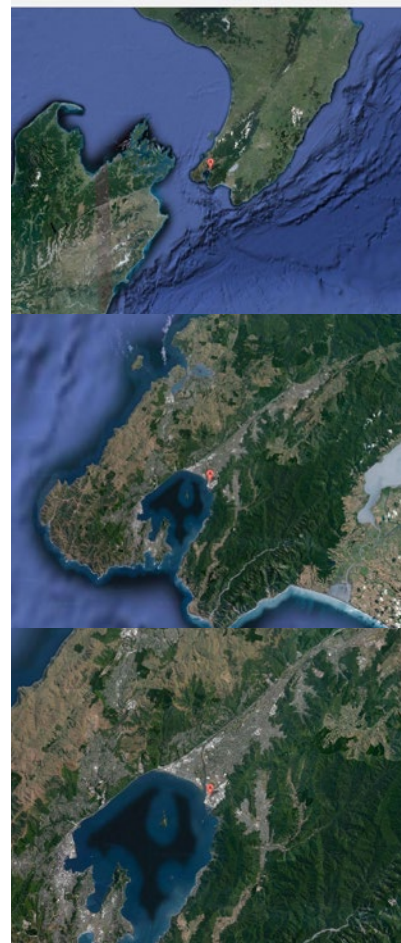


FIGURE 33. Seaview’s contexts.



FIGURE 34. The suburb of Seaview.



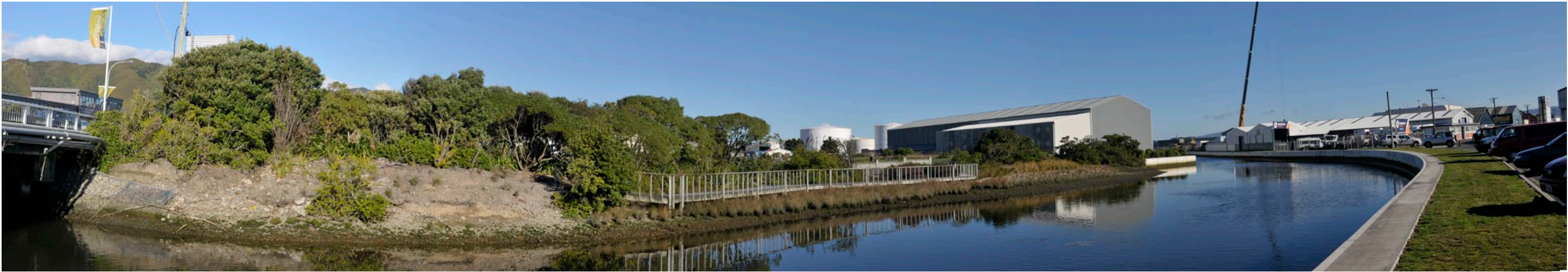


FIGURE 37. Panorama of the Waiwhetu Stream.



FIGURE 38. Panorama of the petrochemical sites adjacent to the waterfront.



FIGURE 35. Stormwater drains and recent planting adjacent to the Waiwhetu.

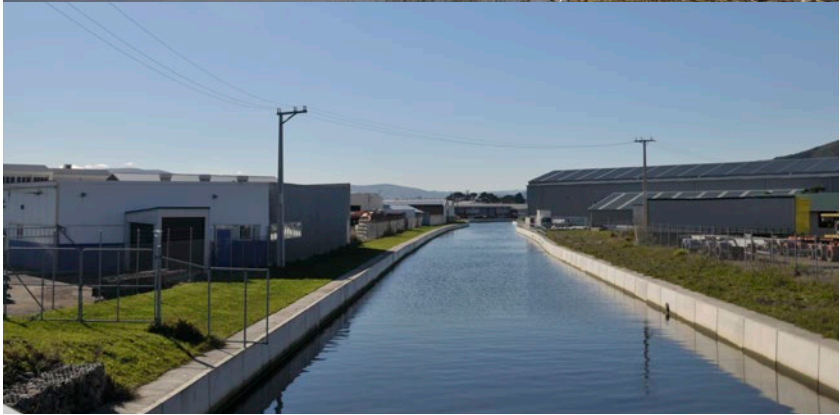


FIGURE 36. Channeling near to the Waiwhetu's mouth.



FIGURE 39. Erosion along the waterfront, and the debris used as a seawall.



FIGURE 40. Waterfront erosion, and the waterfront itself.



5.2 Mapping Analysis

My design process started by gathering site data from a number of sources:

- GIS data sources from the Hutt City Council, Greater Wellington Regional Council, and various national government departments.
- On site observations, by way of notes and geo-tagged photographs.
- History books and photographs from the Wellington City Archives.
- Technical reports prepared by environmental consultants for the Greater Wellington Regional Council.

This data was collated, classified, and imported into Rhinoceros. From there my investigation focused on hydrology, with an eye to the interesting infrastructures present on the site: the river, the waterfront, the stormwater system, and the oil tanks. These were mapped (Figure 41). In each of these axonometrics, the stormwater systems are shown along with a figure/ground of the land and water bodies, along with cadastral packages, building footprints, and a 50mx50m grid reference.

Early versions of this map attempted to use standard techniques to represent land use and program, such as colouring or filling the outlines of buildings and cadastral packages. This approach was complicated by the large amount and variety of plot sizes, and the presence of mixed use programs. Instead, the Field pattern was used to produce a grid that represents a spectrum of land use in terms of programme and intensity of use. Although Seaview is relatively monocultural in its land use, the resulting map helped distinguish key areas of intensity where multiple programmes meet, as well as the overall patterns of land use. Of particular note in this map is the lack of recreational programmes around the river and waterfront, and that most stormwater pipes drain directly to the river and ocean.

With an eye to erosion and flooding as two key site issues that required resolving, I moved to a new round of mapping. The base layer for the map aimed to depict a more precise overview of flooding risk by using the Topos pattern to interpolate new contours from the terrain mesh. These new contours were then used to create overlapping opaque layers that form a gradient from least flood-prone to most flood-prone. I also used the Field pattern to depict the shifting salinity of the river based upon measurements from a technical report. Finally the Flow pattern was applied to the waterfront and riverbanks to investigate drainage issues (Figure 42).

The Field visualisation of the salinity gradient worked well as a graphic. However, because the data was simple and continuous, the representation did not reveal much beyond an overview of a simple overall trend. The flooding susceptibility gradients worked slightly better, providing both an overall trend, as well as several points of local salience. The Flow tracings were better still. Because each of the paths generated by the script were depicted at a low opacity, their overlaps created a cumulative effect in which common water drainage paths stood out. Moreover the actual results of the simulation were surprising: while the water generally drained towards the river or ocean, the common drainage direction was actually away from the river in many cases.

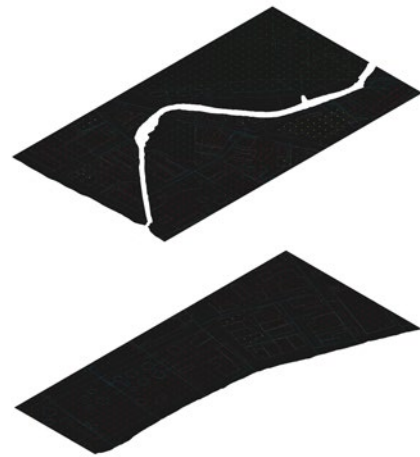


FIGURE 41. Hydrological infrastructures in Seaview. Larger versions on pages 47-50.

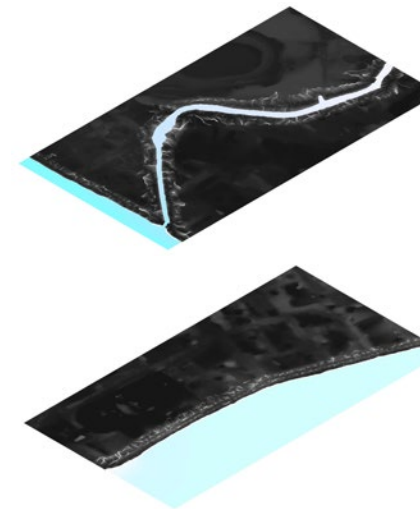


FIGURE 42. Water flows, flood gradients, and the salinity distribution. Larger versions on page 51-54.

Following this I moved to a final mapping exercise which would investigate the existing ecological conditions (Figure 43). Using a contour base map, I applied the Fields pattern to the information that identified plant populations alongside the river and the ocean. Here only a single colour was used, with the saturation and size of the circle representing the density of plants.<sup>197</sup> A second use of the fields pattern was set up to map the ecotone of the areas of non-commercial land adjacent to the river and ocean. Here the red-yellow gradient represents the degree of water saturation in the soil, based upon topographic data and technical surveys. Yellow areas would be expected to be ideal locations for semi-aquatic plants, while red locations would be suitable for fully terrestrial species.

One final element was added to this map using the Sections and Levels patterns. Combining the Levels pattern with technical data on flood events, I created a series of thresholds for each of the different water levels that could be expected along the riverbanks and waterfront. These covered events such as high tide, low tide, a 10 year flood, and a 50 year flood. I then set up section lines that followed the path of the river and the ocean. When combined, these generated a series of 200 sections that show the terrain of the river and ocean set against various water levels. The section lines are indicated on the map (Figure 49) and the section themselves presented separately (Figure 50). This sectional analysis was very useful, both in gaining a better understanding of the river and water front terrain, and in identifying particular areas that were vulnerable to flooding. While the overall results of the ecological mapping were not surprising, the fineness of scale proved useful in creating the subsequent design proposal.

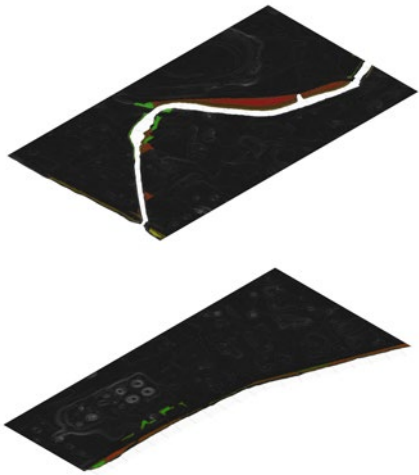


FIGURE 43. Topography, section lines, and the distribution of plants. Larger versions on page 55-58.

5.3 The Initial Design Proposal

The data gathered from the mapping analysis was used to design a planting and grading plan for both the riverbanks and the waterfront.

The first move was to create a planting plan using the Seeds pattern that would match particular species to the conditions identified in the salinity and ecotone maps. I created two Excel spreadsheets that represented an aquatic and a terrestrial planting palette. Once potential areas for planting had been identified, I then created a Grasshopper definition that linked the Field and Seeds patterns so that the salinity and soil saturation analysis could be fed into the algorithm that places each of the plants. This process is represented in two diagrams (Figure 51). This allowed me to automatically generate an overall planting plan for the riverbanks and waterfront, and to easily tweak the species used along with their relative distributions and the areas specified for planting (Figures 52-55). The species were chosen for a combination of ecological and aesthetic reasons, taking into account the present and future state of the plants that the Seeds pattern depicted.

The design developed further by introducing a number of new elements:

- Artificial wetlands would be created on the waterfront and riverbanks, creating stormwater filters and enhancing amenity.

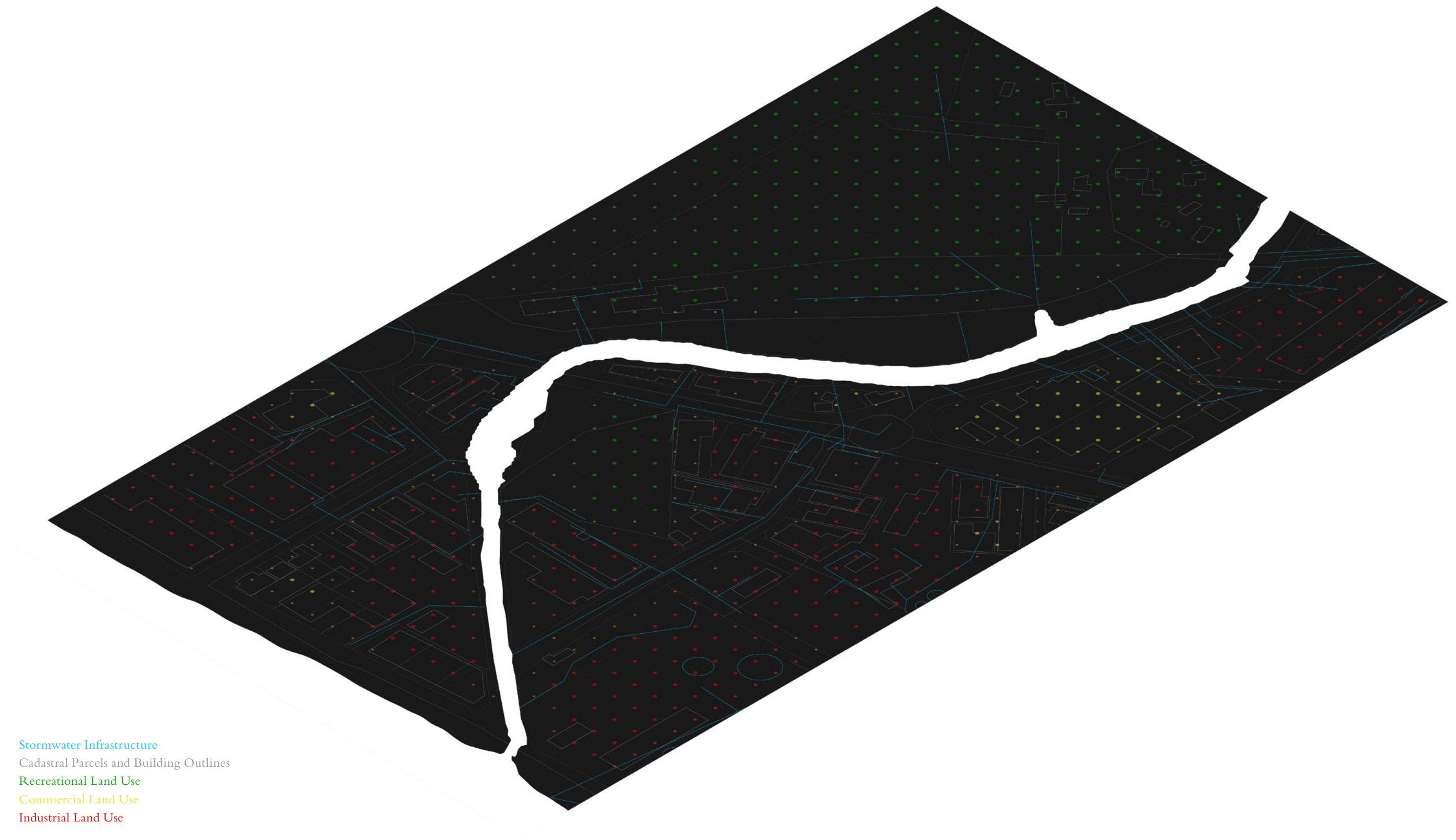
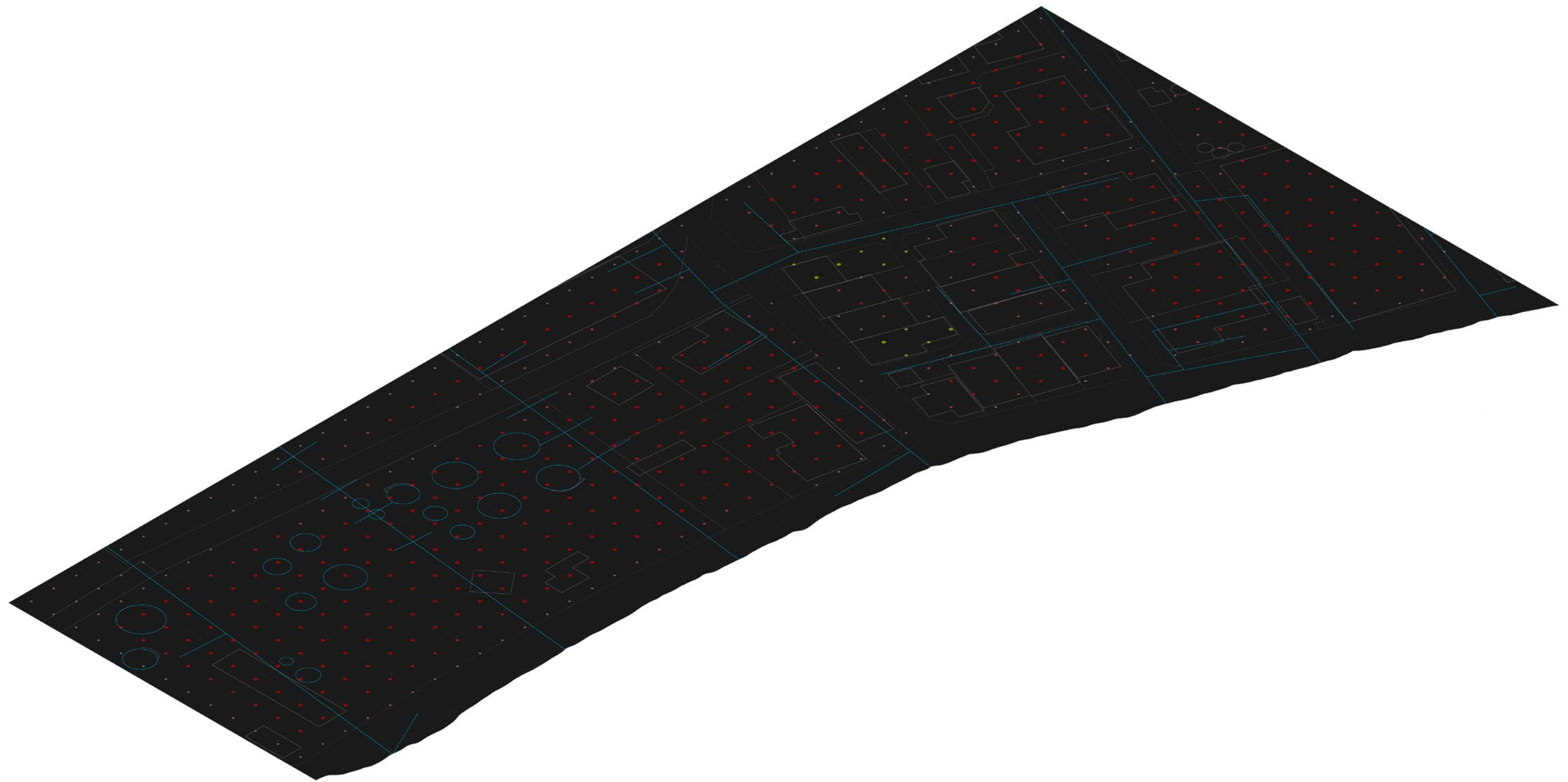


FIGURE 44. Hydrological infrastructures and land use surrounding the Waiwhetu stream.





Stormwater Infrastructure  
 Cadastral Parcels and Building Outlines  
 Recreational Land Use  
 Commercial Land Use  
 Industrial Land Use

FIGURE 45. Hydrological infrastructures and land use surrounding the Waiwhetu stream.

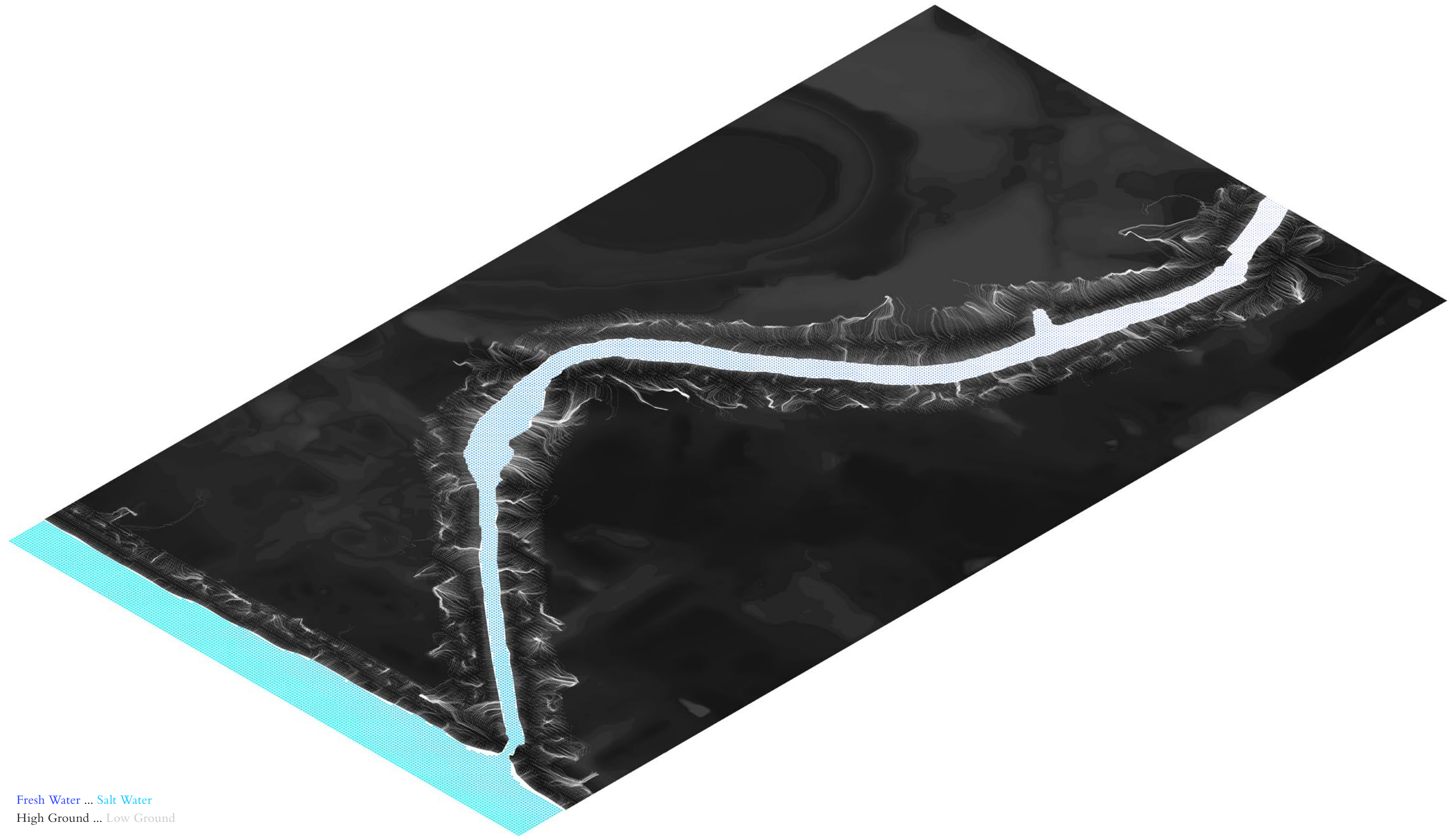
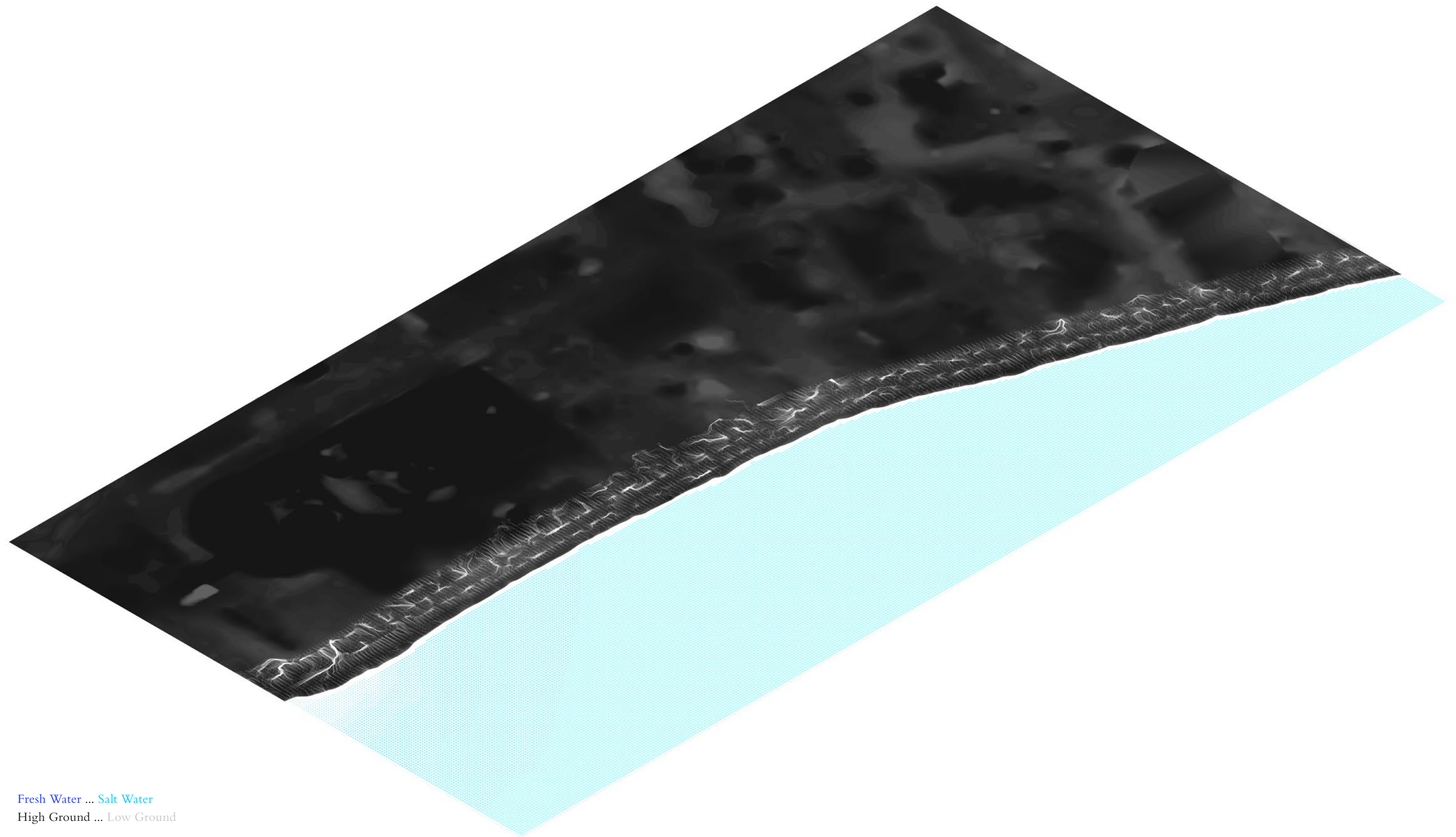


FIGURE 46. Water flows, flood gradients, and the salinity distribution surrounding the Waiwhetu.





Fresh Water ... Salt Water  
High Ground ... Low Ground

FIGURE 47. Water flows, flood gradients, and the salinity distribution surrounding the Waterfront.



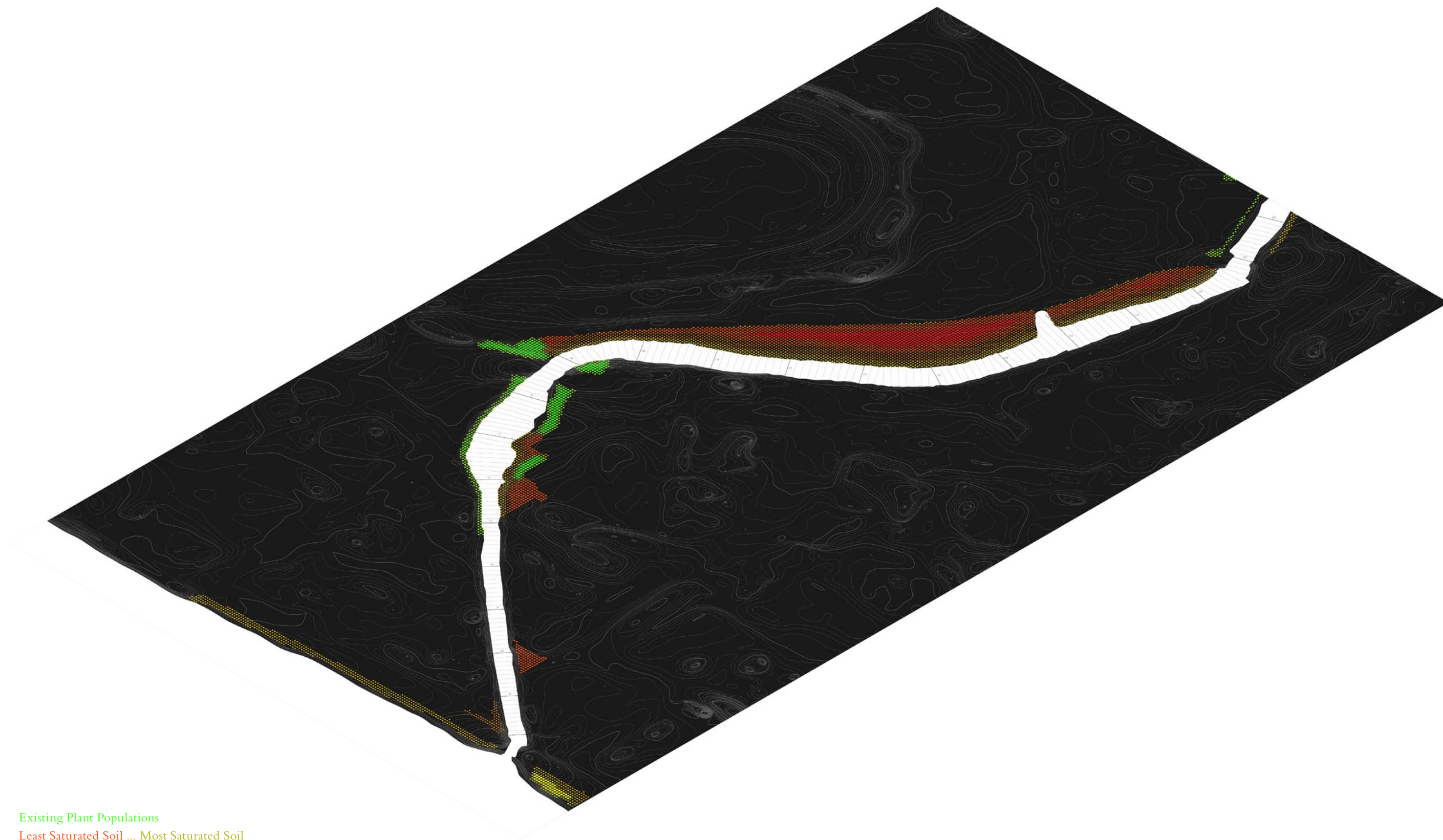
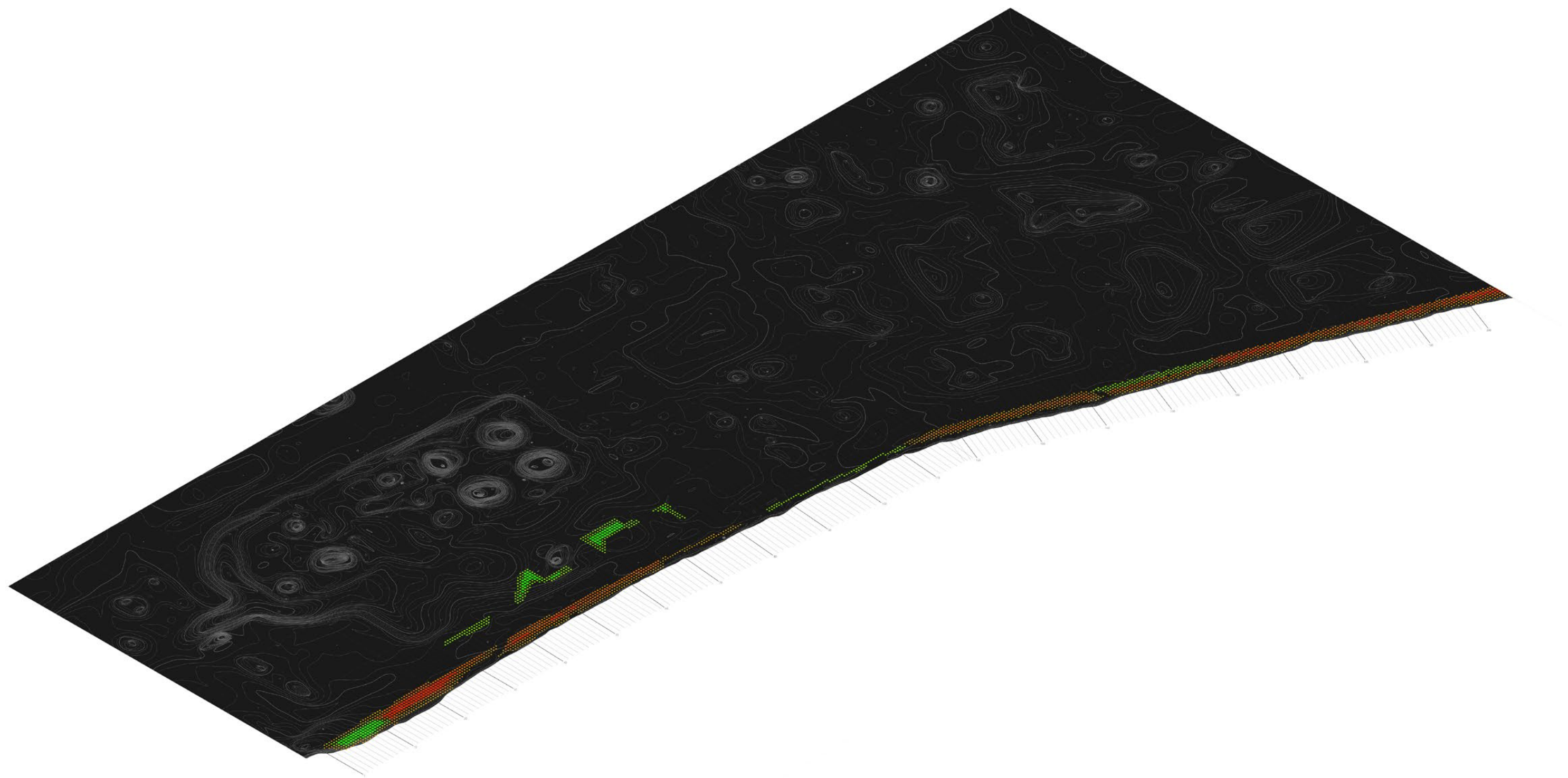


FIGURE 48. Topography, section lines, bank conditions, and the distribution of plants surrounding the Waiwhetu.



Section Lines  
 Existing Plant Populations  
 Least Saturated Soil ... Most Saturated Soil

FIGURE 49. Water flows, flood gradients, and the salinity distribution surrounding the Waterfront.



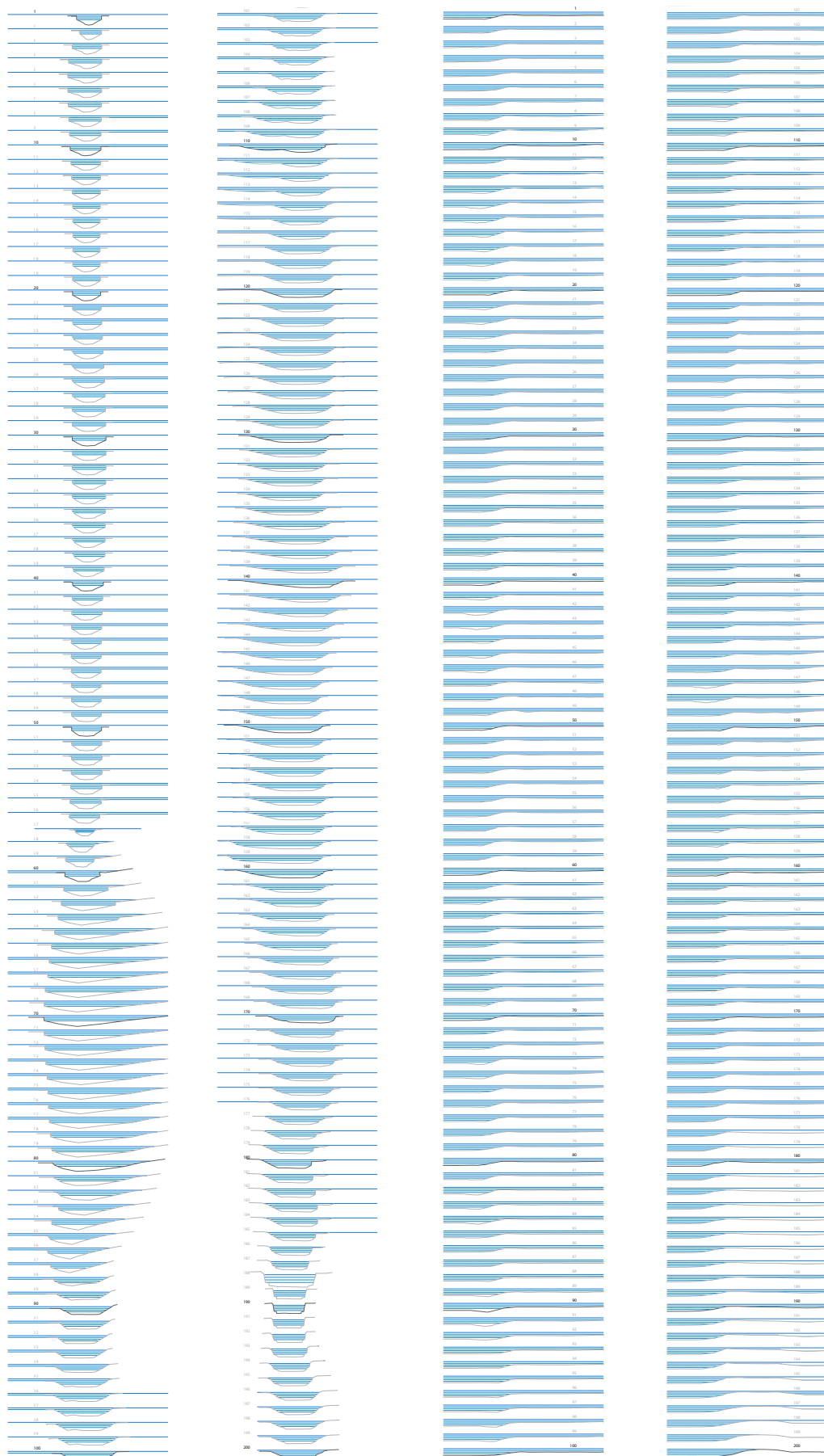
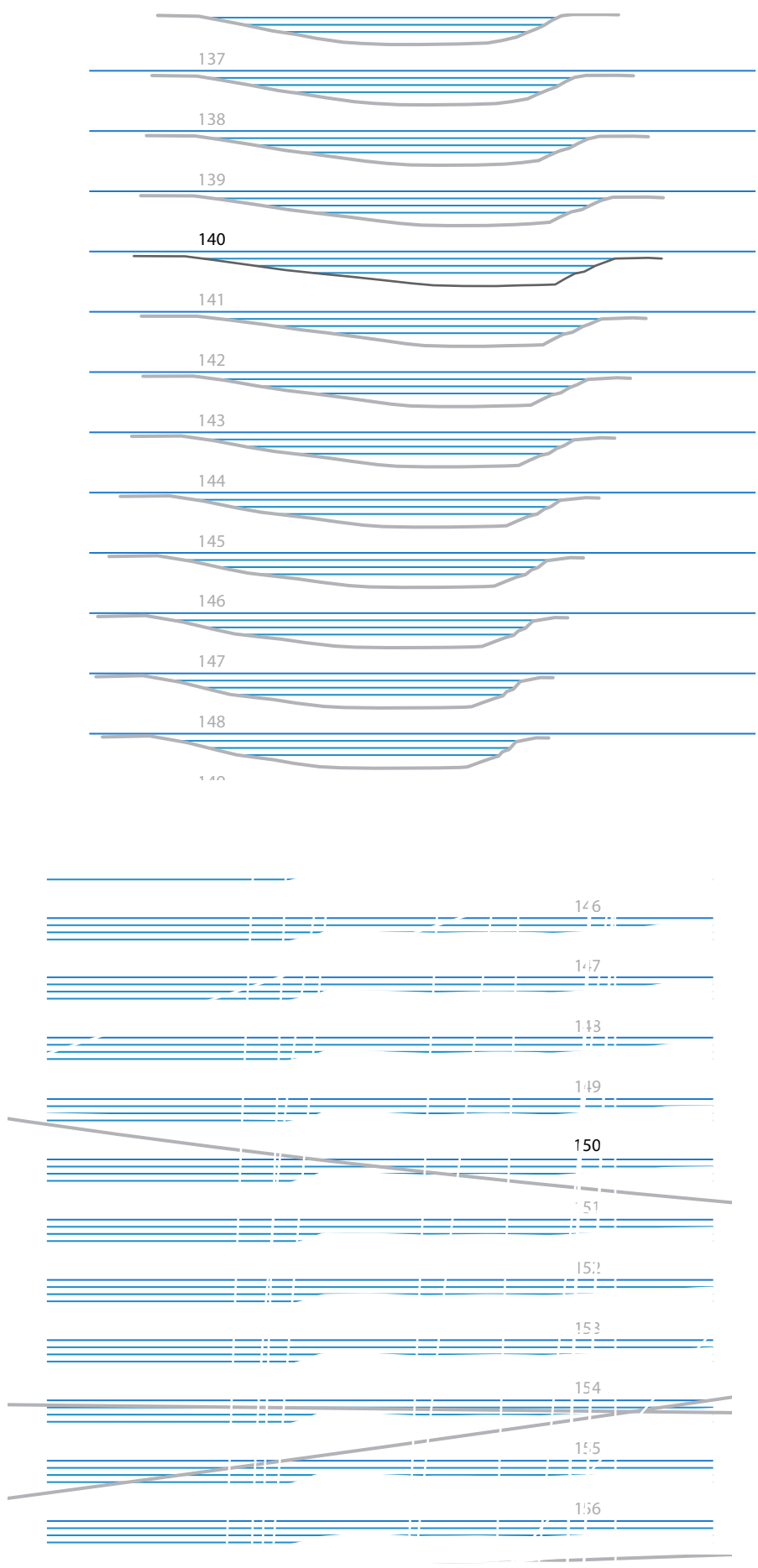


FIGURE 50. Serial sections, showing flooding heights, through the Waiwhetu and the waterfront.



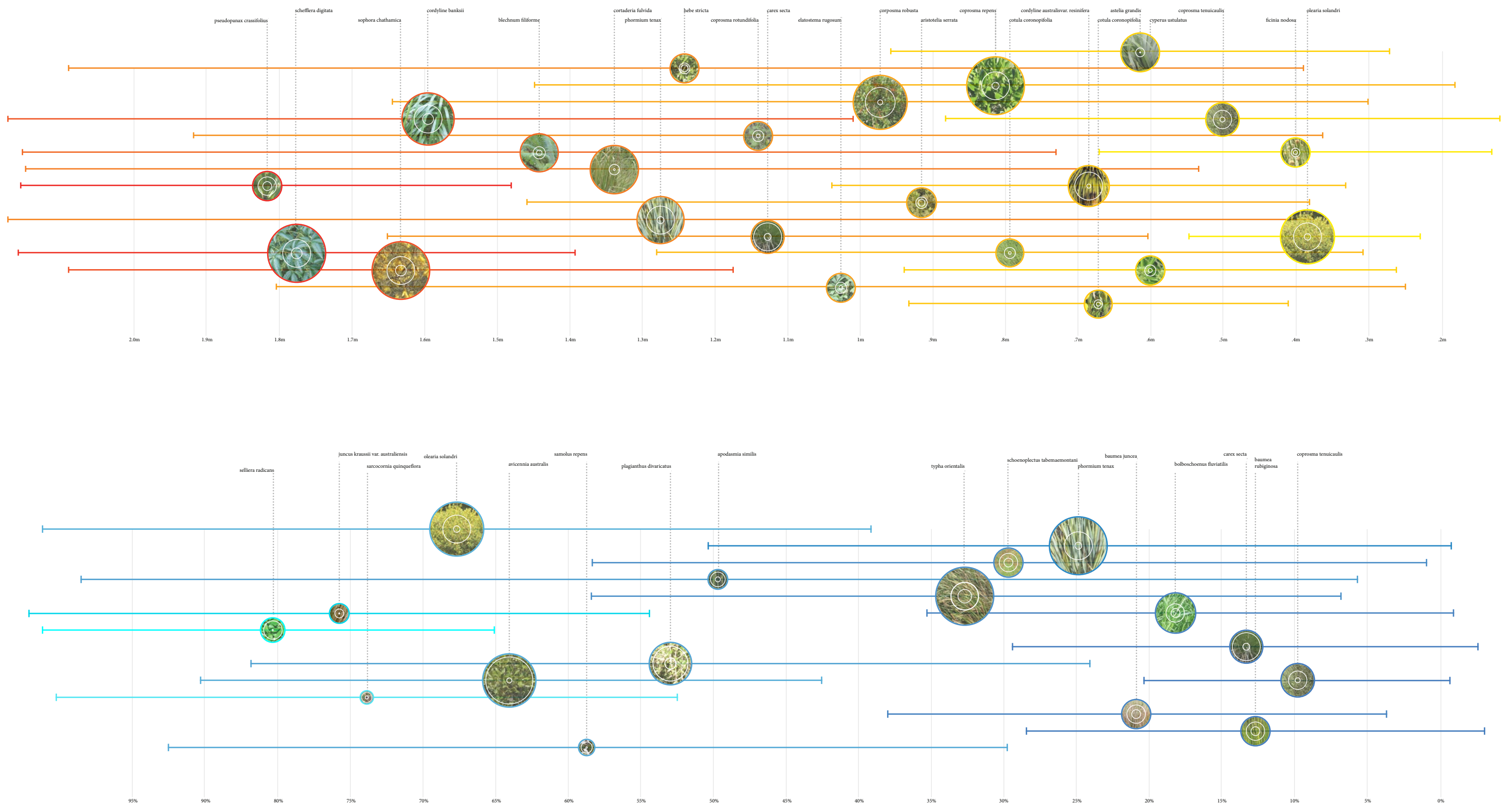


FIGURE 51. Diagram detailing the automated process of assigning particular plant species to a particular location. The top diagram shows the range of soil tolerances and species for the riverbank plantings, while the bottom diagram shows the range of salinity tolerances for the river plantings.



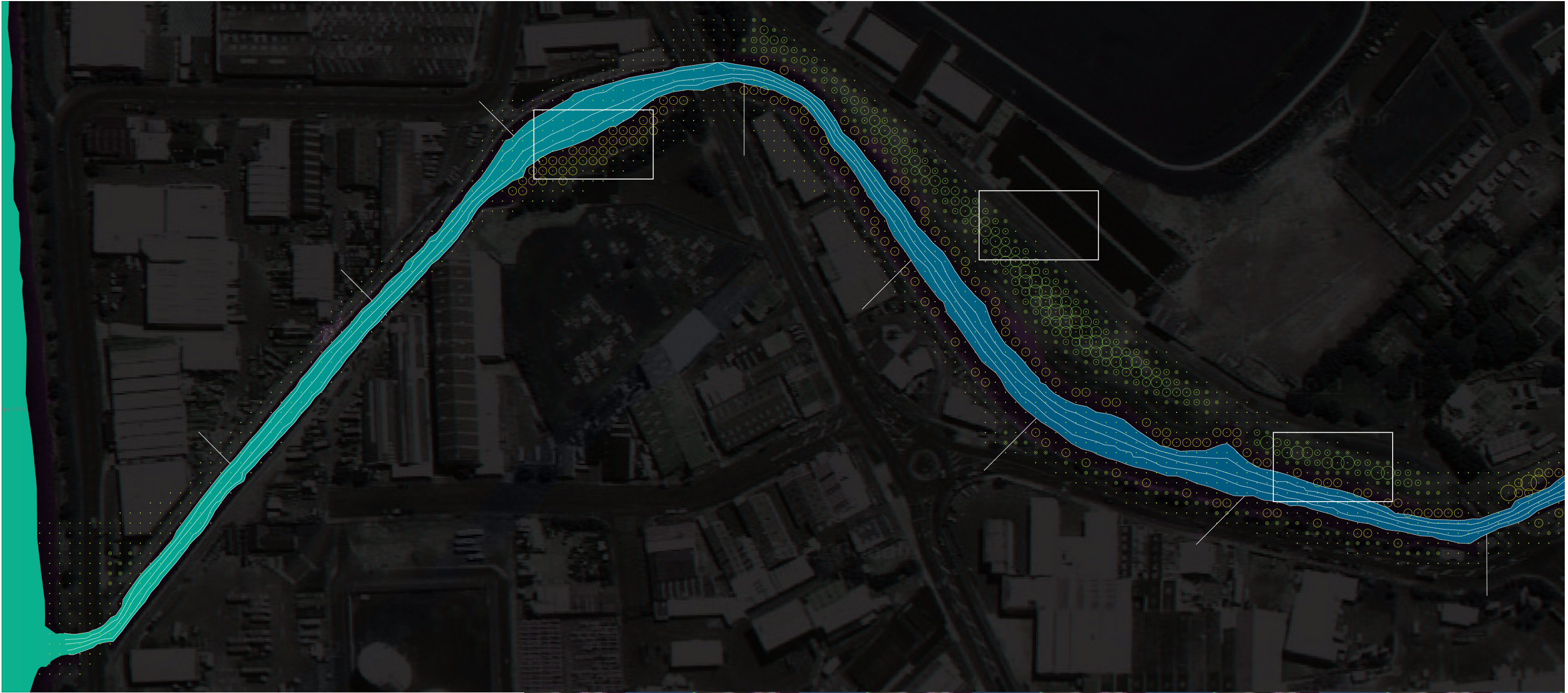


FIGURE 52. Plan of the automatically generated planting plan for the Waiwhetu river.

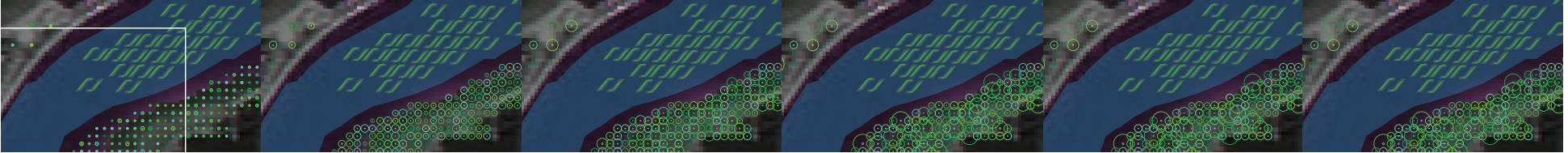


FIGURE 53. A portion of the planting plan showing the simulated growth of plants over 5-year increments.



FIGURE 54. A portion of the planting plan showing the simulated growth of plants over 5-year increments.

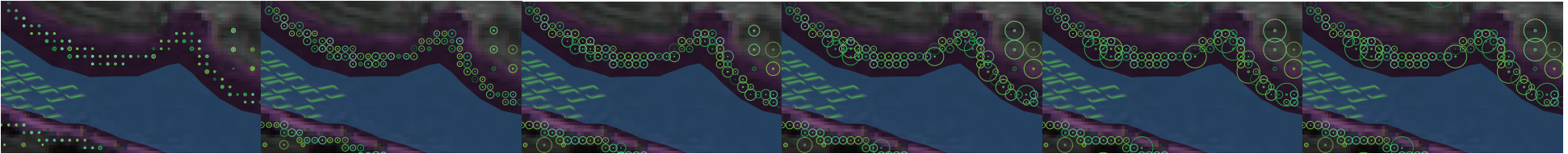


FIGURE 55. A portion of the planting plan showing the simulated growth of plants over 5-year increments.



- The terrain would be regraded to provide better planting opportunities and to help address erosion. New retaining walls and green mats would be introduced, and the current stormwall debris on the waterfront would be shifted outwards.
- Various amenity elements, such as walking tracks, cycling paths, sculptures, and seating, would be added to help meet the goals of the Vision Seaview Gracefield proposal.

This approach was questioned by myself and in my design reviews. While the use of patterns to generate this proposal created some novel visualisations and enabled design iterations to be produced rapidly, the final outcome did not demonstrate that these tools could create a interesting and unique design. The planting plan, and the further additions proposed above represented a modest addition to the site that roughly matched the 2030 Vision. The lack of strong formal interventions into the site meant that the design fell short of demonstrating that both the ‘architectural’ and ‘landscape architectural’ approaches of landscape urbanism could be synthesised. This approach did not make for a particularly compelling demonstration of the potential of computational tools as a way to expand what is possible in landscape architectural design. While it was unclear whether the generative potential of these tools needed to be demonstrated through a novel design, I decided to try it.

#### 5.4 Subsequent Design Proposal

The next iteration of the design proposal concentrated on a larger area of land than the narrow confines of the riverbanks and waterfront. This new area is a large plot currently used for petrochemical processing and storage. For the purposes of this proposal it was assumed that this site would be vacated by its current owner and redeveloped by the Council as a mix of public spaces and new subdivisions available for commercial or residential use (Figures 57–58).

This new choice in site was guided by the decision to dramatically change the course of the Waiwhetu River itself. Given that the channelling of the river was responsible for a dramatic decrease in the biodiversity and amenity, the new design would abandon the channel and redirect the river’s path. The challenge here would be to plan a viable redirection without increasing flood risk while also creating a compelling new public space. In this way the goals of the Vision 2030 document would be achieved, but in a more ambitious and interesting manner. Changing the site would also solve a key flaw in the original proposal: that the banks besides the river channel are too small to accommodate a pedestrian link through to the waterfront.

With this in mind, the overarching concept for the revised design was to create a space that mixes ecological functions with a public park. A series of more extensive formal operations could create a scaffold for new programs and a new ecosystem:

- Implementing regrading and artificial structures to create wetland ecosystems attuned to the path of the new river.

- Reusing the iconic petrochemical vats as follies to house sculpture, and surrounded by water basins.
- A slightly regrading of the site to allow the public spaces to be used as a containment area for flooding protection.

To develop these, the use of the patterns was expanded and augmented with more form-based computational techniques.

The first part of the new design was to redirect the river southward through a new wetlands area. Here, the new terrain and wetlands were designed using a branching division technique, whereby a self-dividing line was overlaid onto a surface to create a tree-like series of paths. For the river, this meant that the new wetlands area would begin with a series of small mound shapes that eventually transition into larger mounds, before becoming solid ground (Figure 58). The differences between the elevation and surface area of the mounds creates an ecotone that spans from a submerged wetland, to a semi-submerged wetland, to a littoral patch, to a forest. The size, quantity, and length of each of the branches was able to be easily controlled through the Grasshopper definition, which made tweaking and modifying the layout of the wetland simple (Figures 62–65). This surface division was linked to the Levels pattern to gauge the different soil conditions on and around each mound, which was then linked to the Seeds pattern that would generate plants tailored to the water and soil conditions of each location (Figures 60–61). This was particularly complex given the changes in water volume that occurred as the size of the branches increased; the need to maximise the surface area of the mounds; and the need to ensure the river would actually flow along the path.

The combination of the branching system, the Levels pattern, and the Seeds pattern created a novel design for an artificial wetland system that could rapidly transition along an ecotone by controlling a complex mix of aquatic and terrestrial factors. Although this ecosystem would have benefits in terms of ameliorating flooding and stormwater pollution, it was designed for aesthetic effect; creating a walkway for the public that dramatically demonstrates a compressed ecotone that represents the larger ecotone which was historically present in the area.

Once this initial stretch of river-wetland becomes mostly-forest, the river is diverted through a series of large pipes underneath the public space. A portion of the river is day lighted along the main paths to maintain a visual connection to the water. While passing under the public space the pipes feed the water pools that surround each of the storage tanks. Again, the purpose for this is largely aesthetic: under-grounding the river and controlling where it is revealed creates a more dramatic experience and opens up a larger area of public space. This space is then used by the tank/sculpture follies, while the grassland area that can be used for sports and other activities. The entire area is below the grade of the surrounding land so that it can function as flood basin for the surrounding stormwater system to drain to. The Levels and Flow patterns were key in testing this, allowing me to consider the capacity of the site to accommodate flood water, and to ensure that the flood water would

flow there quickly. The Seeds pattern was also used to create a planting plan for the banks that surround this basin, where large forest trees were selected to help enclose the park from the surrounding industrial area and to complement the skyline that is currently dominated by the tanks.

The final stage of the design unearths the river by running it down two separate channels that lead to the ocean. These act as the inverse of the initial wetland stretch. The same surface division strategy is applied to transition from the parkland space out into the water, with the ecotone transitioning from the forest to the ocean. The levels pattern was linked to the planting pattern to match the salinity tolerance of the plants to the level of tidal flushing that occurs as the grade reduces.

Throughout the design process, the mapping techniques used in the initial site analysis were continuously applied. A final series of maps were produced (Figure A) that represent the new design.

This design revision represents a much more ambitious attempt at improving amenity and ecology than the original design, although the two plans are not mutually exclusive. The subsequent design is a much less idealised form of landscape urbanist strategies, in that it relies on a relatively intensive, one-off intervention into the site rather than a slowly-implemented strategy. That said, phased elements are incorporated in terms of the planting plans and the ability for the site to adapt to seawater flushing and floods. There is potential to use a more phased strategy to implement the formal interventions, and to account for the impacts of erosion and human occupation. However, the current patterns did not have these capabilities, and developing this functionality would have been time consuming. Similarly, it would have been interesting to follow the design process further in order to resolve more precisely the technical challenges and details of the design, but I was sceptical that this would have revealed much about the computational tools in question, aside from their ability to accelerate detailing.

Although the site diverges slightly from the process-based strategies of landscape urbanism that were critiqued in Chapter 4, it does not replace them with the architectonic strategies of an architectural landscape urbanism. It does heavily intervene into the site by creating a new path for the river, but because this is a change to landform it is open to being able to affect, and be affected by, future changes to programmatic uses and ecological conditions. While this falls short of finding a grand bargain between the two schools of landscape urbanism, it follows the largely expected practical realities that — especially in medium-scale sites — formal interventions into the landscape are a practical response. Perhaps a subsequent design revision, or selecting a different site, would create a more fertile ground to explicitly investigate how these two schools of landscape urbanism could be merged.

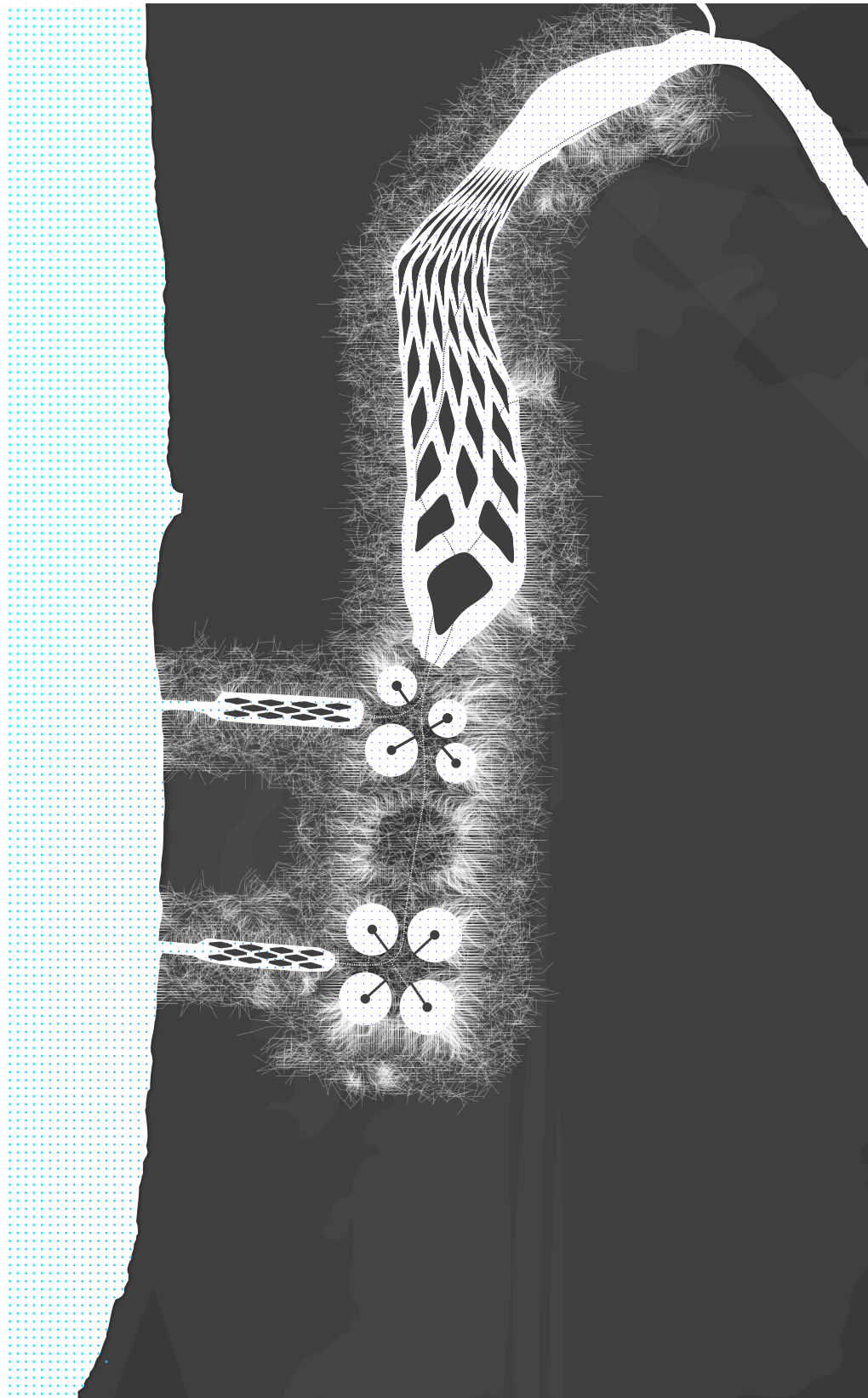
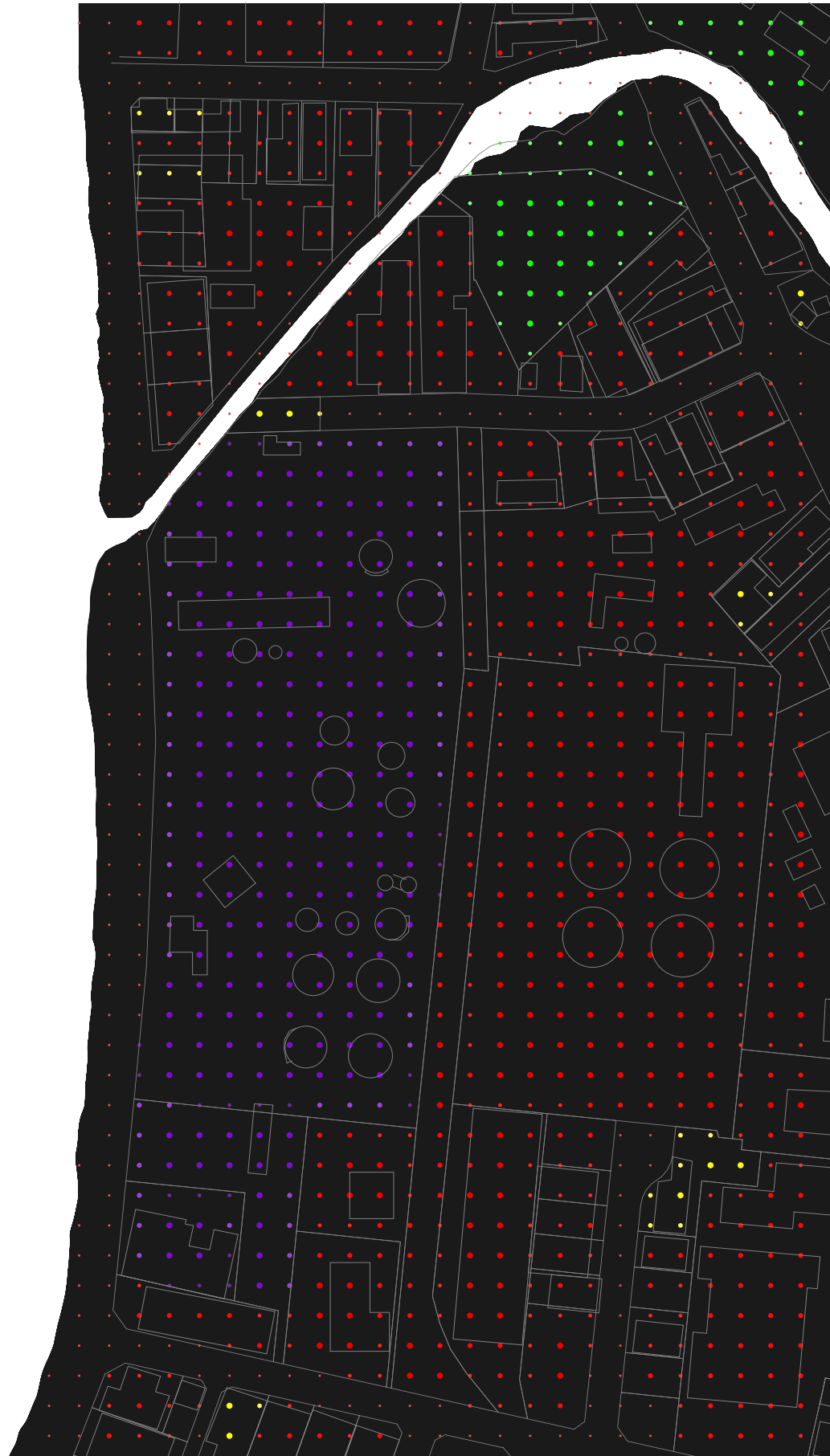


FIGURE 56. Salinity gradient, and water flows, and water's path for the second design iteration.



Commercial Land Use  
 Recreational Land Use  
 Petrochemical Land Use  
 Industrial Land Use

FIGURE 57. Current land use.



Commercial Land Use  
 Recreational Land Use  
 Petrochemical Land Use  
 Industrial Land Use

FIGURE 58. Proposed new land use.



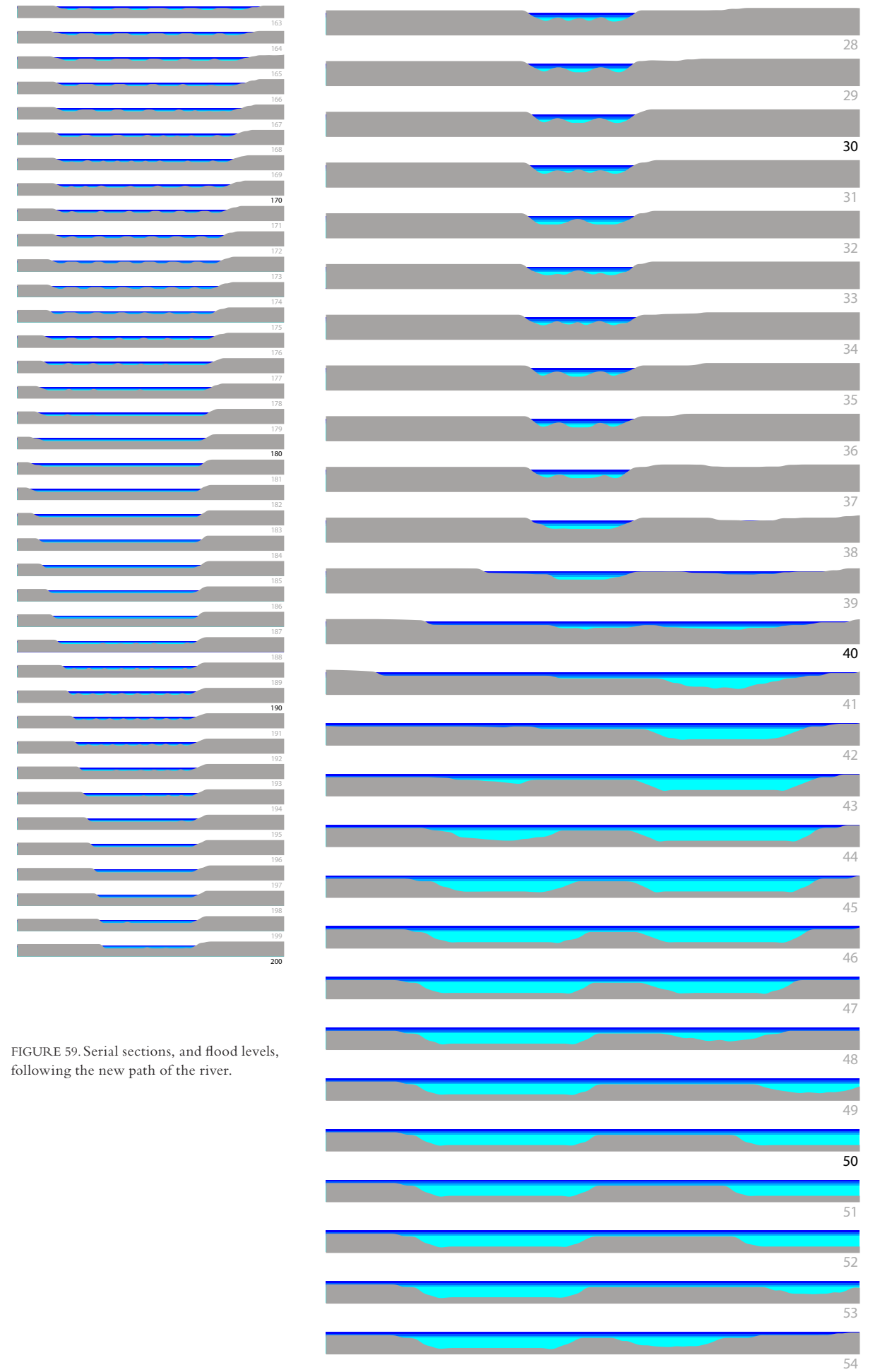
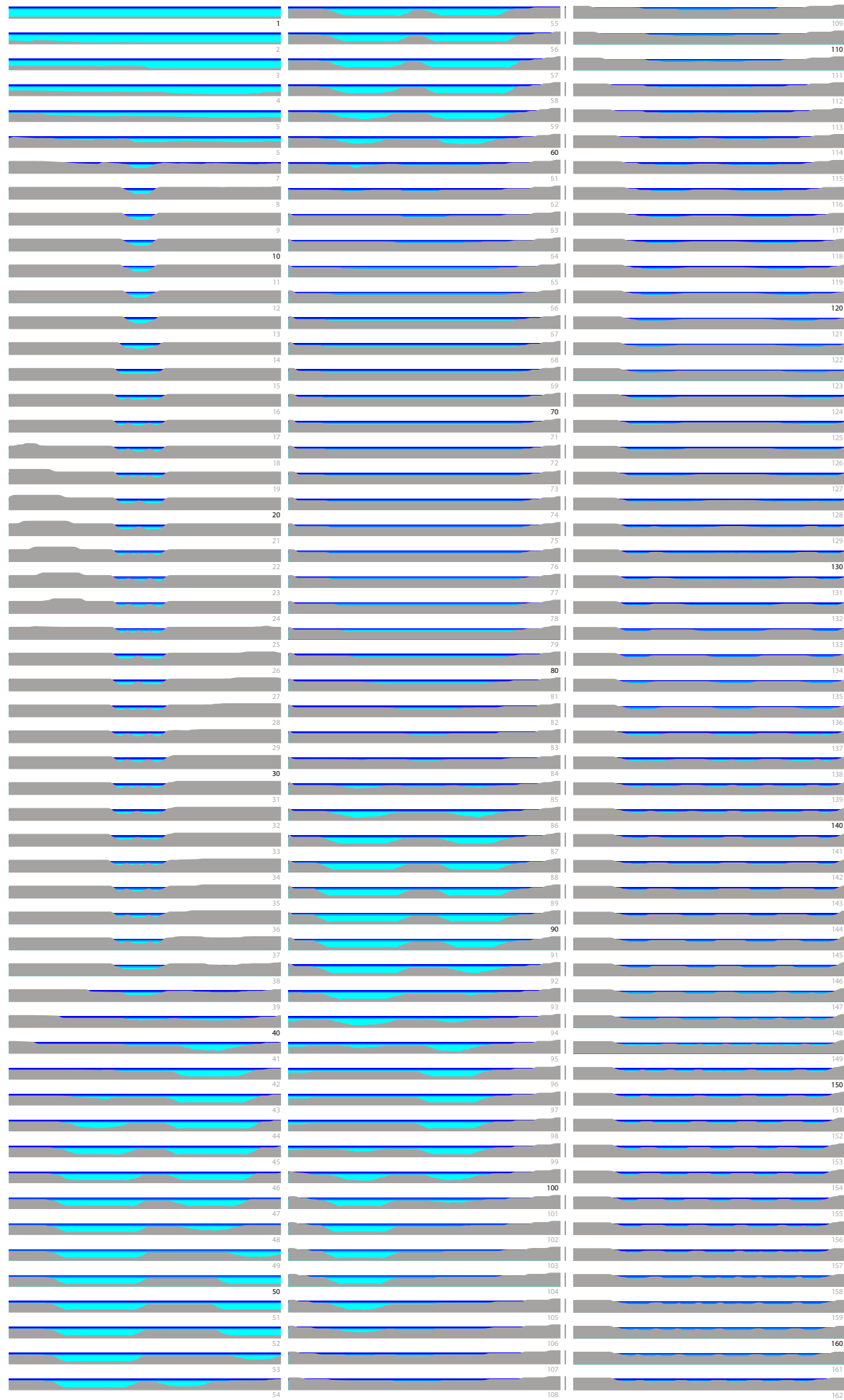


FIGURE 59. Serial sections, and flood levels, following the new path of the river.

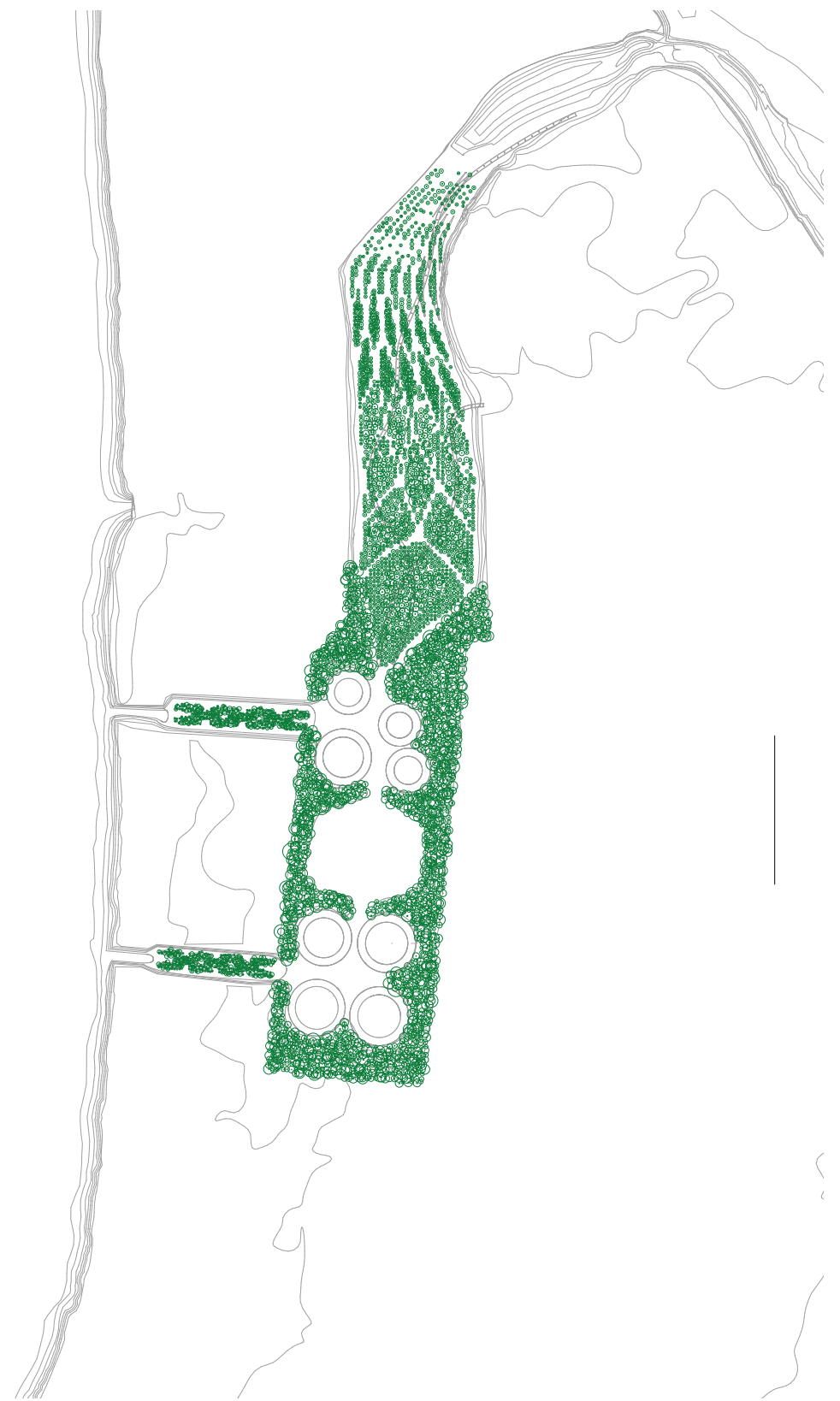


FIGURE 60. Planting plan for the second design iteration, simulated at 1, 10, and 15 years after completion.



Species Name	Common Name	Indigenous Name	Years to Maturity	Death Rate (%)	Relative Distribution	Spacing Radius (mm)	Initial Foliage Radius (mm)	Mature Foliage Radius (mm)	Foliage Variance (%)	Initial Trunk Radius (mm)	Mature Trunk Radius (mm)	Trunk Variance (%)	Initial Height (mm)	Mature Height (mm)	Height Variance (%)
<i>apodasmia similis</i>	jointed wire rush	Oioi	5	1	1	1200	300	1000	15	100	200	15	300	1250	25
<i>aristotelia serrata</i>		makomako	30	1	2	5500	500	4000	20	200	400	20	714	6000	20
<i>avicennia australis</i> var. <i>resinifera</i>	mangrove		30	1	1	750	300	1000	15	100	400	15	500	1500	15
<i>baumea rubiginosa</i>	orange nut sedge		25	1	2	600	500	750	15	100	200	15	200	1000	15
<i>carex Secta</i>	costal tree daisy		30	1	2	600	375	850	15	10	50	200	14	1200	20
<i>coprosma rotundifolia</i>		mikimiki	25	1	2	1600	125	1000	20	25	75	14	300	2000	15
<i>cordyline banksii</i>		ti ngahere	25	1	2	1600	428	2000	20	150	300	20	1000	2000	20
<i>corposma robusta</i>		karamu	25	1	1	5500	214	2000	20	428	400	20	714	4000	20
<i>cortaderia fulvida</i>		toetoe	5	1	2	1600	150	2000	10	50	100	10	250	1500	10
<i>elatostema rugosum</i>		parataniwha	6	1	2	1600	125	1000	20	50	75	14	50	500	15
<i>hebe stricta</i>		koromiko	15	1	2	1600	100	1250	20	50	150	20	200	1750	20
<i>myoporum laetum</i>		ngaio	28	1	1	5500	375	3000	10	50	200	14	500	5000	15
<i>olearia solandri</i>	costal tree daisy		30	1	2	750	375	500	10	50	200	14	400	1000	15
<i>phormium tenax</i>		harekeke	30	1	2	1600	375	2500	20	125	250	20	300	2000	20
<i>phormium tenax</i>		harekeke	25	1	2	1600	375	2000	20	125	250	20	300	2000	20
<i>pittosporum tenuifolium</i>		kohuhu	35	1	2	5500	500	4000	10	50	200	14	600	6000	15
<i>plagianthus divaricatus</i>	salt marsh	maakaka	6	1	1	750	125	1000	20	50	75	14	200	2000	15
<i>pneumatopteris pennigera</i>	lime fern		10	1	2	1600	500	1500	15	100	200	15	300	1500	25
<i>pseudopanax crassifolius</i>	lancewood		10	1	2	5500	285	2500	20	100	200	20	857	6000	20
<i>rhopalostylis sapida</i>			50	1	1	5500	250	3000	15	50	50	15	500	5000	20
<i>samolus repens</i>	maakoako		10	1	3	750	100	450	15	50	100	15	50	200	20
<i>sarcocornia quinqueflora</i>	glasswort	ureure	15	1	2	750	100	400	15	50	50	15	100	200	20
<i>schefflera digitata</i>	pate		35	1	1	5500	571	3000	20	200	400	20	714	5000	20
<i>schoenoplectus tabernaemontani</i>	lake clubrush	kapungawha	30	1	3	600	500	1000	15	100	200	15	400	1200	25
<i>selliera radicans</i>		remuremu	15	1	2	750	200	1000	15	25	25	15	100	300	20
<i>sophora chathamica</i>		kowhai	30	1	2	5500	571	2000	20	200	400	20	857	6000	20
<i>typha orientalis</i>	rush	raupo	30	1	3	600	750	750	15	100	200	15	400	1200	25

FIGURE 61. Plant species, and their attributes, used for the planting plan.

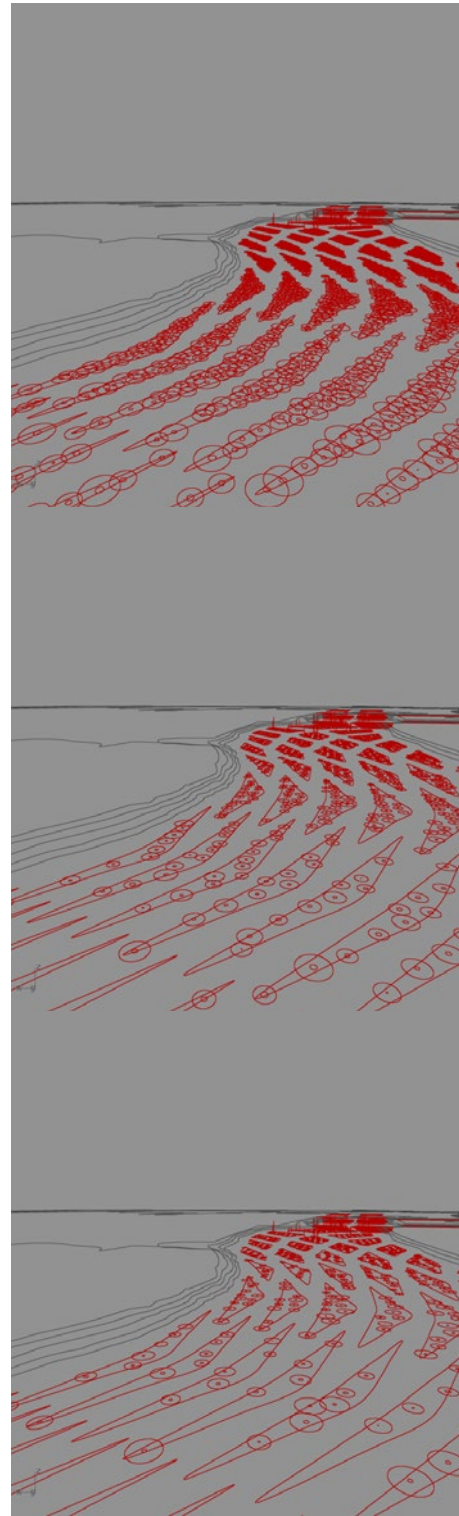
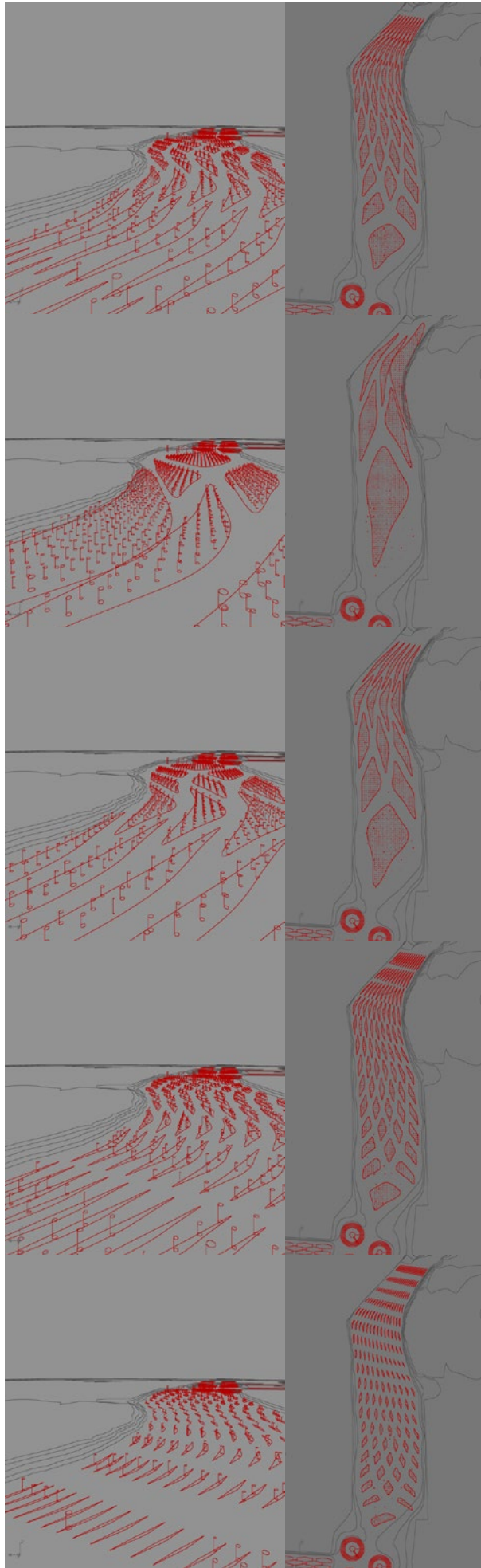


FIGURE 62. Adjusting the density and distributions of plants for the wetlands area.

FIGURE 63. Changing the branching structure of the wetlands area.

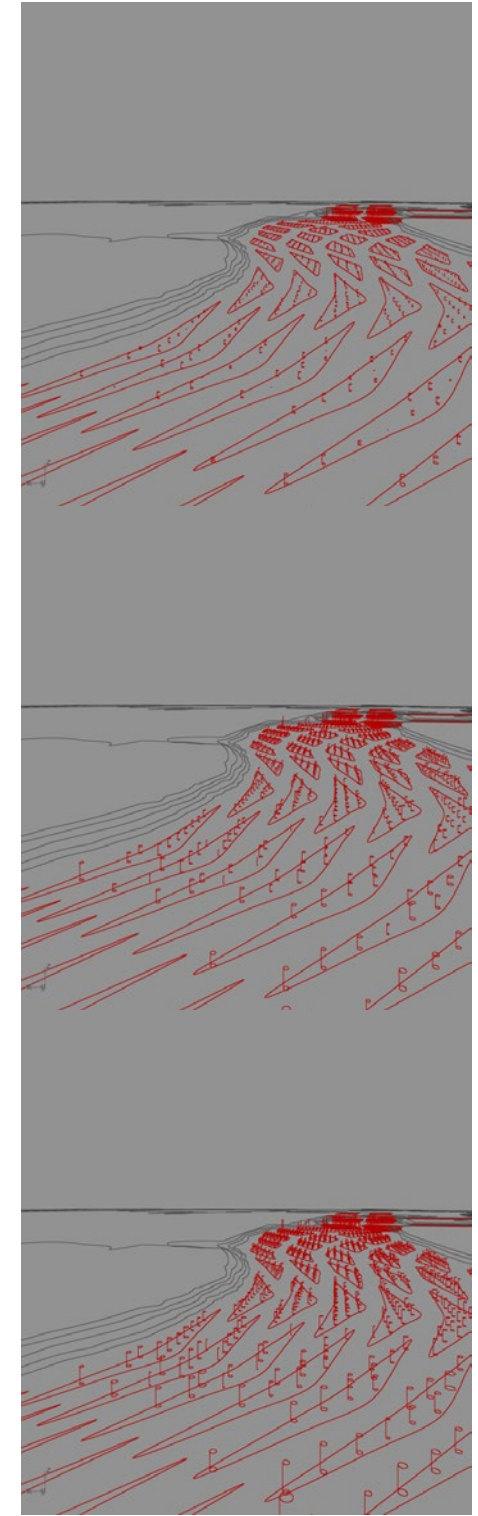
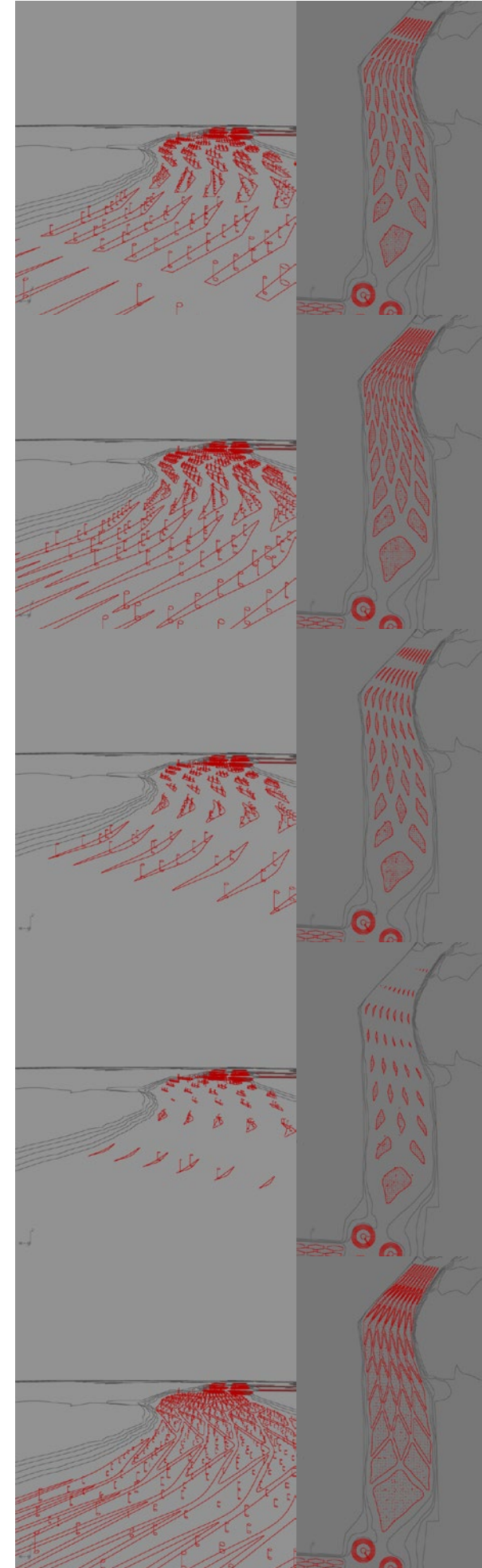


FIGURE 64. Simulating the height-growth of plants in the wetlands area.

FIGURE 65. Changing the size and shape of the wetland 'islands.'



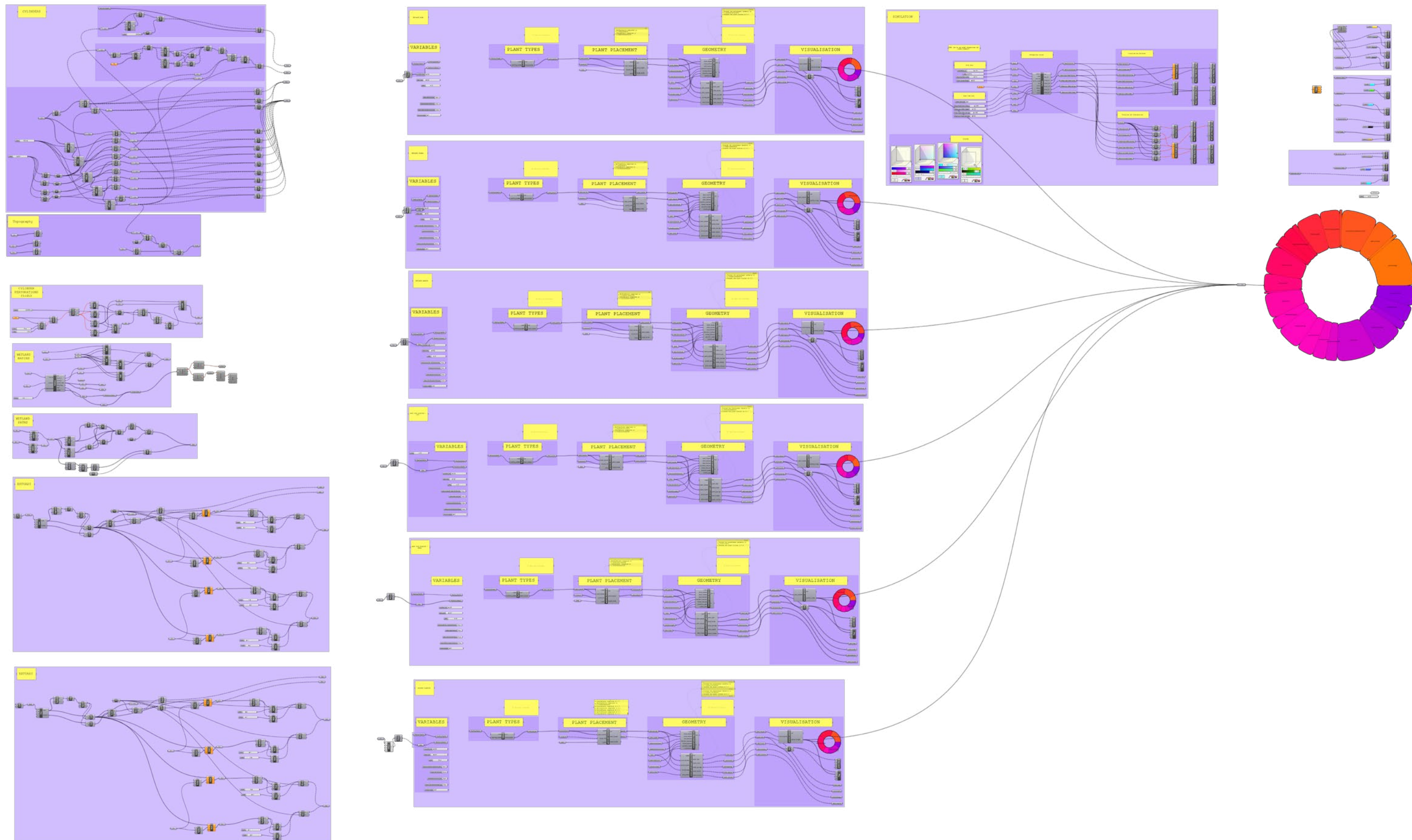


FIGURE 66. The complete Grasshopper definition used to generate the design.

## 5.5 Evaluation of Patterns

As discussed in the previous chapters, the patterns were designed to provide techniques for analysing landscape systems that could be used in a generate capacity to create designs that work with the dynamic nature of these systems. Various precedents, such as the work of the AA's landscape urbanism programme, demonstrate the use of computational techniques to analyse landscapes, but there are relatively few instances which demonstrate the same techniques being used to create designs that are architectural, rather than landscape architectural.

Overall it was found that focusing on analytic techniques was an extremely useful approach. By having a carefully delimited functionality, my techniques can easily interface with traditional computational techniques that define form in a more prescriptive manner. When used in conjunction, the two form a feedback loop that enables conceptual elasticity and technical rigour by allowing new design to be easily judged by the designer against subjective criteria while also being judged by the computer against objective criteria. Moreover, the use of temporal simulations allows a design to be easily projected into the future.

That said, each particular pattern had a different use-case and differing levels of utility within the design process:

- Seeds was useful in its ability to rapidly iterate through both the technical and aesthetic aspects of planting plans. By linking the pattern to site data (using the Fields pattern) or site forms (using common Grasshopper components) it was possible to start designing planting plans much earlier in the design process. This was particularly helpful when designing the wetland area in the second design iteration, where the precise combination of surface divisions and plant typologies underwent many rapid iterations before the final options were settled. The complexity of performing just one iteration using traditional CAD techniques would have been extreme. Automated tools exist for visualising plants within 3D-modeling programs, but they focus solely on aesthetic concerns and do not offer the flexibility that operating within Grasshopper affords.
- Fields is useful in creating analytical maps in which there is a large amount of complex information, particularly when the data overlaps or changes subtly. This pattern is a kind of update to the common layer-cake style of mapping in that it synthesises multiple variables into a single representation. However, it does so in a way that leads to an enhanced understanding of the data at a global and local level, whereas traditional cartographic techniques are less precise. It better synthesises the power of the synoptic map with the productivity of working in the middle, as discussed in Chapter 5. While the use of circular graphic symbols to represent quantitative information is not novel, the computational implementation of this method makes this style of mapping accessible and simplifies creating complex multi-variate representations.

- The Levels and Sections patterns offered utility, but not novelty; they mainly accelerated what was previously done manually. That said, Levels was useful in judging relatively simple phenomena such as flooding and sea level rise in relation to a design. It would need further development in order to be account for the criteria (such as water volume and velocity) that define more complex hydrological phenomena.
- Flows was very effective in the initial analysis phase, and appears to be a technique that combines the rigour of specialist hydrology software with the accessibility of common CAD software. That said, after the initial analysis it was not used as a generative tool, being instead run to validate major design decisions. There was unused potential here to have the simulation itself affect topography, rather than being used to merely judge manual changes to topography. Working with a smaller and more topographically-varied site would have been a more effective place to test this capacity.

## 5.6 Limitations in Programming

Using computational techniques to design buildings is hard. Unlike a typical design process, it requires in depth technical knowledge of both programming and geometry. Over the course of my research it became clear that using computational techniques to design landscapes is harder still. The temporal and non-linear nature of landscape systems requires the designer to use advanced programming techniques in order to represent the stochasticity, interactivity, and self-organisation that drive these processes and their formal effects.

Two of my patterns, Flows and Seeds, attempted these more complex programming techniques, and were notably more difficult and time-consuming to create. Due to time constraints, several other patterns that I had imagined, or started, were not finished, such as a pattern looking at the directional flow of water, and simple 'boid' simulations that could be used to represent populations of people or animals.

I do not doubt that advanced simulations of ecological phenomena are possible. Ecologists have been building increasingly advanced implementations of these tools over the past decades; all that needs to be done is for the code to be adapted for use within a 3D-modelling environment. That said, there is a significant challenge in this undertaking, and in using these kinds of simulations in general. Computational models of landscapes introduce a second-order of dynamism because both the model and the medium are dynamic. While this temporal dynamism is extremely productive when designing landscape systems, it introduces a new layer of complexity into the computational design process.

Some factors mitigate this complexity. My pattern-based methodology is designed to enable novice users to use these advanced tools, and modify them in an increasingly advanced manner as their skills increase. In my own use, and in limited testing with other users, this approach has promise. Many of the popular Grasshopper plugins operate in a similar manner, such as Kangaroo Physics, which allows architects to use advanced structural simulations.



In summary, this approach has potential for both experienced and novice users. The evolution of Grasshopper, and other computational design communities, demonstrates that ensuring these tools are easy to pick up is particularly important. If there is to be a distinctly landscape architectural application of computational design, computational tools need to be both effective and accessible. Producing and sharing quality code is key to this aim.

## 5.7 Limitations in Representation

There were several limits encountered throughout the design process. The most notable of these was a challenge in representing 3D models of landscapes. Landscape architecture, like architecture and interior architecture, uses multiple types of representation for different ends. Typically, most work is presented at the end of the design process through representations that detail a design to a reasonably high level of realism and fidelity. When working with 3D modeling software, renders usually perform this role. 3D rendering is relatively rare in landscape architecture. As Roberto Rovira notes:

*“It is often more challenging to achieve realistic visual representation in landscape architecture than in other disciplines, such as architecture and interior design, where the depiction of space generally relies on the accurate representation of objects.”*<sup>198</sup>

Landscape architects typically create Photoshop collages and Illustrator compositions rather than using the semi-automated process of rendering. This is likely due to a kind of ‘uncanny valley’ effect where renderings of nature generally look plastic and fake unless the renderer is particularly skilled. Because the use of manual, rather than semi-automatic, presentation methods is favoured within landscape architecture, modifying a design after presentation graphics have been produced imposes a high cost.

This was problematic for my design case study as it explores the initial stages of the design process where any presentation-quality compositions would become rapidly obsolete. It was even more problematic given that a key part of my investigation was to demonstrate that computational methods can quickly and rapidly produce distinct design iterations. To create presentation quality graphics to display all of these iterations would have been incredibly laborious. Videos documenting these demonstrations would likely have been effective, but did not square well with the printed nature of this document. It is possible that developing better rendering capabilities, or programming a degree of automation in the transition from Rhinoceros to Illustrator/ Photoshop would have increased the quality of the presentation graphics.

The second kind of representational limit was within the design process itself. Working within a 3D modelling program still seems foreign to landscape architecture. More so than architecture, there is a propensity for landscape architects to work with the relative abstractions of plan and section. Because landscapes are often tricky to represent, our discipline seems to rely more on the the designer’s eidetic imagination than in architecture, where geometric representations do a better job at realistically depicting the designed object. I

often decrease the fidelity of a landscape model depending on the design task. There is a distinct lack of affect when working with 3D models that seems to be exacerbated when designing landscapes. It is worrying that this could lead to designs that embody this lack of affectivity.

Increasing the fidelity of textures and models could mitigate this, as could developing better forms of abstraction in the software that better mirror drawn methods.<sup>199</sup> Various new techniques also offer promise. Augmented reality could enable a more spatial interaction between designer and model that combines actual experience and virtual simulation.<sup>200</sup> Rapid prototyping offers a means to quickly fabricate a tactile representation of a digital model.<sup>201</sup> The use of 3D scanning and other techniques may enable the designer to work with tangible materials to produce forms that could be automatically digitised. Investigations into these techniques are beyond the scope of this thesis, but are promising steps towards mitigating some of the problems that arise when using 3D landscape models.

## 5.8 Conclusion

This chapter has described how my patterns, used in conjunction with standard computational design techniques, can enhance the design process in landscape architecture. This was demonstrated by developing two concept designs for the Seaview region.

Through each concept design it was demonstrated that the patterns were effective in improving the mapping process by enabling the designer to better understand complex conditions. These new methods of mapping could then be linked directly into the design process as data sources, enabling the design itself to better adapt to the complexities of landscape systems. In both designs, issues of hydrology and planting were particularly effective examples of the use of computation techniques, allowing a design to be created with more flexibility, more complexity, more fidelity, and more speed than traditional methods.

I started this thesis by asking how computational design could improve landscape architecture. Answers were sought by investigating the computational design process, in theory, and at two levels of practice: the general and the specific.

This chapter will review the results of this investigation, looking at the significance of the results, their relationship to established literature, their limitations, and the opportunities for further research.

6.1 **The Conceptual Investigation**

The theoretical investigation was fruitful. Having investigated architectural theory and landscape architectural theory, I identified areas of commonality that relate to computational design. Architects that use computational tools theorise their design process in terms of diagrams, systems and processes. It is significant that this mirrors how landscape architecture discusses its own design process, highlighting the affinities between landscapes and computation as mediums. Surprisingly, landscape architecture seems better equipped to understand computational design because it has already embraced variability and complexity within the design process.

This is important because the key barrier to using computational design techniques is cited as a conceptual — not technical — misunderstanding of the medium. A landscape architectural use of computation would create a novel second-order dynamism within the design process, as both the designer’s tools *and* the designed object are both dynamic systems. This isomorphism between process and product greatly increases our ability to harness the complexities of landscapes within the design process. In particular, this synthesis could solve several current problems:

- Buildings created using computational tools are often criticised for formal extravagance. Performative criteria are increasingly used, but there remains a lack of contextual engagement within computational design.<sup>202</sup> A focus upon landscape architectural problems could demonstrate how computational tools can productively engage with contextual conditions, pushing both disciplines forward.
- Landscape urbanist strategies are often criticised as being too assertive about how ecological processes will act. An increased ability to represent, simulate, and design these processes using computational tools could create more precise representations of such processes.
- The landscape architectural design process struggles with the complexities involved in representing the scale, dynamism, and temporal nature of landscape phenomena. A greater focus on the virtual conditions of these systems, rather than their actual appearance, would benefit designers because

it allows them greater agency over how these systems operate. Because computational tools operate through generative rules, they are the perfect method to make the complexities of landscape systems become manageable and malleable within the design process.

More public case studies — in theory and in practice — are needed to test this proposed common ground between philosophy, architectural theory, computer science, and landscape architecture. It would be a great aid if the landscape architects currently using computation in practice were more open about when and how they use it. Even more useful would be if an online community could form to discuss and distribute computational techniques for landscape architecture. The presently architecture-dominated Grasshopper community shows this is both possible and productive.

6.2 **The Design Investigation**

The goals and criticisms identified in the theoretical investigation were tested by creating a series of computational ‘patterns,’ and by applying them to two case studies. In developing the patterns, the overarching question was whether landscape phenomena could be simulated computationally, and whether this knowledge could be made accessible to others.

My patterns suggest that landscape systems can be modeled using computational methods. That said, doing so is often an arduous undertaking, and it remains unclear what limits there are to this method. For example, advanced simulations of water flow could prove too difficult to implement within standard 3D modeling software, or too computationally taxing to use easily within the design process.

Because programming is typically a difficult and novel experience for designers, it is important that the computational techniques are easily used and customised. The use of a pattern methodology itself, and their hybrid implementation in both Grasshopper and Python, helps with this aim, but a full quantitative study would be needed to confirm this.

In creating my designs I examined how the patterns could improve the design process and its results. I demonstrated that the patterns were able to produce analyses that accelerated and exceeded the capabilities of traditional representational techniques. Moreover, when linking the results of the patterns to standard computational techniques, the two formed a generative process in which interventions into the site could be measured by, and derived from, performative criteria measures over various time periods. This was most notable when designing the primary wetlands area, where geometric, hydrological, and ecological factors were integrated using



computational techniques. This integration allowed the complex interactions between landscape systems to be explored quickly and easily, creating a more considered and precise design. It is significant that this process could not have been easily repeated using traditional design techniques.

Further empirical studies of computational design and landscape architecture could investigate other types of landscape systems and test computational techniques across a greater range of design scenarios. This could include a greater variety of sites, as well as other aspects of a design, such as the ongoing effects of a programmatic intervention or improvements to the detailing process.

### 6.3 **Conclusions**

The combination of computation and landscape architecture is potent. While computation may first appear to be an alien force, the laws of digital processors and landscape processes share a natural affinity.

For landscape architects, computation offers a medium in tune with their message: that landscapes are a dynamic entities which demand an open-ended and indeterminate approach to the design process. This dictate is best tackled using computational tools that are uniquely capable of conceptually expressing, and technically resolving, the complexities of landscape systems.

This is the promise of a computational landscape architecture. My research, and that of others, has identified the great potential of this new field, suggesting that its emergence is merely a question of when and how.

## 7 Appendix A: Process Philosophy and Real Virtualities

The spectre of Gilles Deleuze has hung over architecture and landscape architecture since the early 1990s.<sup>203,204,205,206</sup> In particular, his emphasis on a process-driven understanding of the world remains a key presence in computational architecture and landscape urbanism.

Previously, I have detailed that the concepts developed by Deleuze have been instrumental in the development of landscape urbanism and computational architecture. I have also discussed that work of Deleuze can be used to guide how these two fields should be combined. This appendix details several concepts that set up this synthesis. In particular I present a number of concepts that define Deleuze’s ontological views.<sup>207,208</sup> These are the crux of Deleuze’s relevance to design as they directly influence our understanding of the environment and the design process.

### 7.1 The Problem of Universals

We can classify a philosopher’s ontological position by determining to what extent their ontology depends on the human mind.<sup>209</sup> At one end of the scale are the most human-dependent ontologies that define only the act of perception itself as real: that there is no existence outside of the linguistic and conceptual frameworks that classify, explain, and thus *are*, the information that we receive from the senses.<sup>210</sup> At the opposite end of the scale are philosophers who posit that the world exists completely apart from our experiences of it: that all phenomena exist regardless of whether they are observable or unobservable by the human mind.<sup>211</sup>

The work of Gilles Deleuze is located in the latter camp. His ontology defines an inalienably material reality in which phenomena are composed of concrete conditions.<sup>212</sup> This rejection of the anthropocentric perspective places him in stark contrast to other contemporary philosophers who typically view language as the cornerstone of reality.<sup>213</sup> Because of this distinct difference, the ontology of Deleuze is commonly considered to be his most important project.<sup>214</sup> Operating simultaneously as the subject of, and *modus operandi* for, all of his work, it was the consistent application of his unique metaphysic to a great diversity of subjects that made his research so interesting.<sup>215</sup>

Several key distinctions demarcate the work of Deleuze from the other philosophers who also reject the importance of perception in reality.

The first is the split between materialist and idealist. Other non-anthropocentric ontologies may understand reality as defined by ‘essences’ that classify, distinguish, and thus define phenomena. A prominent example being Plato’s theory of forms, wherein phenomena are mere shadows of an essential quality that transcends each particular material object, experience, or expression.

To illustrate, in observing a dog, it is defined by an essential condition of ‘Dog-ness’ to which the observed creature belongs. This essential concept of Dog is then used to identify and correlate distinct entities (the specimen, Spot) within a whole (the species, Dog). In this way, there are two distinct and dichotomous entities at play; a particular instance of a four-legged-domesticated-mammal, and the eternal archetype of Dogs.

This dichotomy between object and essence is found throughout the history of philosophy, establishing ‘the problem of universals’ as one of a key question in metaphysics. Deleuze’s answer to this question is that the object/essence dichotomy should be abolished. He unifies the two by creating an ontological model where the differences between categories (Cat or Dog) are resolved through understanding the properties of each category as a dynamic process, rather than as properties that derive from language, eternal qualities, or any other abstract concept.

To illustrate this distinction, when observing a dog, it is defined by an overlap between categories: there is a specimen that is also an instance of species. To simplify, we can understand both conditions as contingent upon the interactions between biology and the environment.<sup>216</sup> The category of Dog is defined as a unique species that has arisen due to particular evolutionary processes that occur between generations of animals and their environment. Thus the condition of ‘Dog-ness’ is defined as a roughly consistent set of genes that result from this lineage. Spot is individual who shares this species-specific genetic identity, but is also an individual entity that arises out of unique interaction between genes and their environment-dependent expression. The only difference between these two entities is the temporal scale at which each operates — millions of years for the one, a dozen or so for the other.

Deleuze’s processual definition of properties, and the removal of essences, creates a ‘flat’ or ‘monist’ ontology where the distinction between properties and entities collapses. There ceases to be separate ontological categories for the distinct entity of Spot and the distinct category of Dog. Both the individual instance and the categorical property are defined in the same manner: through the unfolding of a dynamic process. Here entities can only ever be seen as the sum of other entities, and the identity of any entity lies in its genesis rather than a universal archetype. The differences between entities derive from explanations, not descriptions.

This raises a few questions. How are these ‘dynamic processes’ not synonymous with essential categories? Is this not just a shifting of universal properties from the end-result on to the conditions of creation? To answer this Deleuze presents the concept of the ‘multiplicity:’ an explicit and operative conception of how his dynamic processes function. It makes it clear that these processes are *real* rather than a new form of essentialism.



## 7.2 Intensive Phenomena

The distinction between intensive and extensive phenomena is crucial to Deleuze, and to his conception of the multiplicity. These terms originate in thermodynamics, where they form a binary that classifies properties into two types.

A property is said to be extensive if its magnitude depends on the size of the overall system. This means that if the size of the system is reduced there will be a corresponding change in the value of the property. An object’s volume is extensive — if you slice an apple in two, each piece has half the volume of the original whole.<sup>217</sup> In contrast, the magnitude of an intensive property is completely independent of the size of the overall system.<sup>218</sup> This means that if a portion of the system is removed, the property will remain unchanged. Density is intensive: each slice of the apple is just as dense as the original whole.

Of the entities that we observe in an everyday sense, Deleuze points out that the majority are discrete phenomena, and are thus extensive. For example, countries, people, and objects occupy space in an extensive manner because they have clear thresholds, whereas a gas occupies space in an intensive manner. Deleuze suggests that extensive phenomena mask an underlying intensity. The weather manifests as discrete objects, such as rain or snow, yet it is the gradients of change within the atmosphere that ultimately drive the creation of these phenomena. The underlying intensive properties — temperature, pressure, humidity — are all imperceptible, but nevertheless extremely real.

## 7.3 The Manifold

Deleuze uses the concept of the manifold to detail the role of intensive phenomena in creating difference and diversity.

The manifold forms the basic structure of a multiplicity. It borrows heavily from the mathematical field of differential geometry, a field that investigates non-Euclidean conceptions of space through algebra and calculus. While Euclidean geometry articulates co-ordinates within a metricised matrix, differential geometry uses curved spaces whereby co-ordinate points are articulated solely by their instantaneous rate of change rather than their instantaneous location.<sup>219,220</sup> In this way, differential geometries can be said to be intensive because they are not defined in discrete terms.

This accomplishes two important things. Firstly, differential geometries have no fixed datum, no  $x=0$ ,  $y=0$ ,  $z=0$  against which things are measured. The second is that there is no extrinsic means of co-ordination other than the space itself because there is no higher dimension to measure the current plane against. Together, these two conditions create a geometry that can accommodate an infinite (n) number of planes. A collection of n-dimensional planes is a manifold.<sup>221</sup>

Deleuze then intertwines this notion of the manifold with another mathematical concept: that of *phase space*. Phase space is a method of representation where all the possible states of a system are represented simultaneously. To illustrate, the velocity and displacement of a single mass on a spring can be measured over time, showing that after any initial displacement the acceleration and kinetic energy of the spring eventually reach equilibrium at zero. A phase space diagram would visualise this event by plotting these two variables on a graph, and then tracing these values over time. In this case, a spiral pattern would be created according to the reciprocal, but entropic, oscillations between acceleration, energy, and gravity within the spring system.

Deleuze joins this concept of phase space with that of the manifold. Each property of an object can be conceptualised as a ‘plane of possibility’ in which all the possible values for the property exist in a kind of conceptual graph, with each plane comprised of intensive spectrums rather than discrete values. For example the brightness of a light bulb possesses a plane of possibility that is a gradient spanning Off, Dim, and Bright. A light bulb may also possess planes for its temperature, voltage, colour, physical shape, and current.

Deleuze equates each of these ‘planes of possibilities’ as analogous to the planes of a manifold. Each property of an object forms a single dimension within a manifold space that encapsulates all the possible combinations of values that could be expressed. If we were to imagine a light bulb as a system composed only of brightness and temperature, this would create a manifold of two planes — a two dimensional space that encapsulates every possible combination of these two variables. Deleuze’s manifolds acts as n-dimensional phase spaces that create a ‘matrix of potential’ to represent all the possible properties of an entity.

## 7.4 The Attractor

Attractors morph the shape of a manifold’s state-space so that it is biased towards particular outcomes. In this way, each plane of possibility has a trajectory that manipulates its values towards a predisposed condition. As discussed earlier, the spring system has an attraction towards an equilibrium state of rest, thus its properties tend towards zero once disturbed, representing the spring’s dissipation and absorption of the initial modification. In the case of the light bulb, an attractor would represent a dependency between current, temperature, and brightness so that increases in current also increase the latter values. A more complex example is the process of crystallisation or the formation of bubbles. In both cases, a series of physical-chemical tendencies result in specific formal outcomes, and thus can said to act as attractors that guide the system towards a particular state. During crystallisation, populations of molecules collectively seek to minimise bonding energy by forming direct and rigid structural lattices, such as the cubic forms of salt crystals. With bubbling, the population of molecules seeks to minimise surface tension by minimising surface area, thus creating spherical forms. Following these attractors — minimal surface area or minimal bonding energy — matter is guided toward a singularity: bubbles or crystals respectively.<sup>222</sup> In this way, the ‘bubble-ness’ of bubbles has always been present within its manifold as it

is an inherent and immanent property of the matter itself. In this way matter an entity’s identity can be defined without the need for exterior observation or interference because it is a self-organising system driven by endogenous processes.

Bifurcations are transitions that represent shifts in phase space. As an object passes certain thresholds its guiding attractors change, altering the phase space and therefore the behavioural patterns of the object. The flow patterns of water will shift as temperature increases, changing from a steady pattern, to a defined flow, to a chaotic turbulence as the transition from conduction to convection occurs. Each of these pattern changes is a bifurcation in hydrodynamic behaviour, whereby a new set of attractors defines a new behaviour.

7.5 **The Virtual Multiplicity**

To synthesise all of these concepts, a multiplicity is a system defined by the features of a manifold.<sup>223</sup> Each manifold is comprised of intensive planes of potential, which embody all of the possible states of an entity. These planes are driven by unique sets of attractors and bifurcations that differentiate towards specific expressions. The multiplicity thus “gives form to processes, not to products.”<sup>224</sup>

Returning to the ontological distinction established at the start of this chapter, it is traditional to divide the world into the real and transcendental. Deleuze rewrites this by seeing all entities as created and identified by a distinct process. To prevent his worldview from being divided into the real and the possible, Deleuze conceives of these processes as part of a higher concept — the multiplicity — that contains the current state of an entity within itself because it *is* the inherent processes that define all of entity’s possible expressions.

Instead of the traditional essence-object distinction, Deleuze posits an actual-virtual distinction. Following this, all objects have a virtuality defined by multiplicities, and an actuality that is their instantaneous state in reality as matter. The virtual multiplicity define systems that contain both the present, actual state of an entity as well as a morphogenetic recipe that can generate all possible past and future actualisations. In this way, the reality of an object exists *inside* the virtual, and *prior to* its existence in the actual. The virtual shelters the genetic conditions of the real. Entities are always sectional images of a multiplicity: the actualities of intangible — but ever-immanent — virtual structures.<sup>225</sup>

7.6 **Conclusion**

*“The key to the ontology I defend is the idea that the world is made out of individual entities at different levels of scale, and that each entity is the contingent result of an individuation process.”*<sup>226</sup>

In summary, Deleuze creates a unique worldview through an unrelentingly realist ontology that replaces the eternal transcendence of essences with the immanent virtuality of multiplicities.

The mechanics of this ontological system are significant to the theoretical origins and practices within contemporary architecture and landscape architecture, as well as in understanding computational design strategies.



8 *Appendix B: The History of Seaview*

This appendix provides a brief overview of the history of Seaview.

8.1 **1000: A Prehistory**

Seaview sits to the east of a congruence of two rivers.

Like many of Wellington’s foreshores, Seaview as we know it was actualised through an intersection of complex geological, ecological, tectonic and political forms.

As with any foreshore, the intersection of land and sea exists as a semi-stable equilibrium between the flows of water, wind, and soil. From the ocean come the tides; their action a largely periodic washing in proportion to the fluxes of gravitation fields.

To the west of the rivers lies a waterfront. Each wash of the tides brings deposits of sand; each particle coming to be lifted and pushed by the flows of air that angle inland. These winds lift and push the sand from loose to stable positions as each grain verges towards the basins of attraction formed by floral forms. Once seedlings, these plants grow in tandem with the aggregated soils, their roots reaching ever-downward for water act as binding agents that begin to concretise the sand towards a semi-stable mound. As the pioneering root nodules add nitrogen to the soil, less hardy species come in. This forms a dunal ecosystem.

Back at the twin river mouths, this process never takes root. Against the inland-pushing tides and winds, the seaward-pushing river flows ensure a constant flux. From the upstream valley and surrounding hills come fresh water and sediment, creating an intensive mixture of degrees of salinity, sedimentation, and velocity. As the river flows approach the tides, the water stratifies as the more dense saltwater sinks; the fresh floats before thinning out as it enters the harbour.

As it does, the suspended sediments are deposited on the banks and flats of the river mouth. The flows of mud and silt settle outwards, creating flat forms and shallow pools that are periodically flushed as the river’s reaches and retreats according to its flow. With these deposits comes organic matter that create territories for microbial and floral growth. Such organisms decompose the sediment’s organic matter into compounds that in turn feed the burrowing invertebrates that sift across the surfaces, their bio-matter in turn supporting to higher food chain. Unlike other ecosystems, the steady flow of upstream organic matter enables a constant increase in bio-matter that supports an immense amount of life. This forms a estuarine ecosystem.

8.2 **1250: A Settlement**

Two son’s of Whatonga — a Hawke’s Bay chief — move to the area, giving the larger river the name of *Heretaunga*. These Ngati-Tara inhabited the harbour for eleven generations, before the arrival of the Ngati-Ira from the East coast<sup>227</sup>. A wedding between the chief of the former to the leader of the latter joined the two tribes, with the Ngati-Tara coming to be known as Ngati Ira by the beginning of the 19th century<sup>228</sup>.

*“Seafood formed a staple part of the diet of local Maori and until the early 1940s eel, crayfish and watercress were harvested from the Waiwhetu River.”*<sup>229</sup>

Amongst the many settlements of the tangata whenua in the Hutt Valley, the most relevant to this story was Ohiti: a stockaded village at the mouth of star-reflecting stream, the Waiwhetu. As with the Hutt River, the Waiwhetu ran deep and narrow; embanked by lofty pine trees that began a mile out from the river mouth<sup>230</sup>

8.3 **1800: A Change**

Great Britain was experiencing a severe depression, prompting an interest in colonisation as a mechanism to ameliorate overpopulation and distress.<sup>231</sup>

With the arrival of the colonists came new trade and new technologies, most notably the musket and the potato. These new tools upended the power relationship’s between tribes, sparking the Musket Wars of 1818.

The Ngati Ira’s place in this conflicts was to be conquered by an offshot of a Te Rauparaha-led war party in the early 1800s.<sup>232</sup> At Waiwhetu, the Ngati Mutunga and Ngati Tama fought a decisive battle against the inhabitants<sup>233</sup>. Later the Ngati Awa remained to take possession of the ‘part of the Hutt district about Whaiwhetu’ that had been given to Ngati-Mutunga after the raid.<sup>234</sup>

So, four years later, when the *The Tory* arrived at Petone Beach, the members of the New Zealand company were received by the Ngati Awa chief: Te Puni. The result of this meeting was a written agreement for the purchase of land, of which the terms were characteristically unclear.<sup>235</sup> Half of the value of the purchase was to be found in six cases of twenty muskets each, along with cases of gunpowder, cartridges and other ammunition.<sup>236</sup>

Upon the arrival of the purchase receipt in England, a town plan for “Britannia” was prepared, showing gridded street network surrounding the Hutt river.<sup>237</sup>

Come January 1840 the settlers had arrived upon Pito-one beach and begun the process of clearing the land and constructing houses in the lower valley.

Mere months later, the persistent flooding of the newly-named Hutt River “drove most of the colonists to the southern end of the harbour (Thorndon and Te Aro), while a few remained behind in Petone and Lower Hutt to farm amongst the floods.”<sup>238</sup>

8.4 **1855: An Uplift**

In Seaview, as in the rest of Wellington, The Earthquake actualised a dramatic change in topography.

By the river mouth, Frethey’s Island was raised considerably, leading to its re-purposing as a Hutt Park, and infill that saw it landlocked.<sup>239</sup>

The uplift stranded a boat located upstream at the Willcox ship yard, located in the Waiwhetu near White’s Line. Having previously built a 30-ton vessel amongst others, the business was ruined after the stream was no longer navigable by boat.<sup>240</sup>

This upheaval would later set the stage for Seaview’s emergence as a centre of industry; the uplifted sand and swamp providing new land to be drained and built.

The floods continued as the population continued to grow. Moreover the floods worsened, with the flood of 1858 causing massive damage to life and property. The citizen’s response to this was for each household to construct low barriers around their properties, usually to no effect except to stave off the lower flood waters by diverting them downstream to their neighbours.<sup>241</sup> No central planning committee yet existed to create public works.

After two tremendous floods in 1898, the residence “rose in their wrath,”<sup>242</sup> lobbying the New Zealand Premier for the right to tax themselves so that “they may protect their homes and property.”<sup>243</sup> After collecting taxes from each property according to its vulnerability to flood, a survey of the river was completed, and a tender issue for flood protection works.<sup>244</sup>

These works came in the form of a stopbank on either side of the river; primarily composed of shingle or bundled scrub. Once established, grass could grow atop the banks, forming a defence against the flood scouring.<sup>245</sup>

Works in the subsequent decades removed shingle from the river bed, lowering its bed half a meter by 1915, a further meter by 1931, and a final meter by 1939.

8.5 **1900-Present: Development**

In 1929, rail came to the suburb, creating a key economic catalyst. Initially the Wairarapa-Wellington line was built to connect to the new Hutt Workshops — a site for the repair and manufacture of trains — as well as for access to the racecourse meetings held in Hutt Park. That year Caltex Oil moved in the area and the Point Howard Wharf was created in anticipation of future industries.

These industries came. At present, the Seaview/Gracefield area contains approximately half of the industrial floor space in Lower Hutt and Wellington, and is one of the few areas with special regulations that allow and encourage heavy industrial uses.

The century of industrial use took a toll on the river, which became steadily more polluted. Clean-up efforts began in the late 1990’s, which culminated in major civil works starting in 2009. Many of the contaminants in the river’s bank’s were excavated, others left in place but covered with caps. The river itself was dredged, an effort aimed at both removing pollution and decreasing flood risk. Finally, the river’s end was straightened and channelled to further decrease flood risk.

8.6 **Conclusion**

The recent history of the Waiwhetu is a century-long bifurcation of the river’s morphology from a dynamic intensive multiplicity into an extensive singularity. This charts a neutering of process; a river transformed from an ecosystem to an exposed pipe.



- 1 Mark Burry and Andrew Maher, “The Parametric Bridge,” in *Acadia* 22, 2003.
- 2 Kostas Terzidis, *Algorithmic Architecture* (Routledge, 2006) 57.
- 3 John Frazer, *An Evolutionary Architecture* (Architectural Association Publications, 1995) 55.
- 4 Terzidis, *Algorithmic Architecture* 54.
- 5 William Mitchell, “Roll Over Euclid,” in *Frank Gehry, Architect*, 2003 354.
- 6 Timothy Lenoir and Casey Alt, “Flow, Process, Fold,” in *Science, Metaphor, and Architecture*, 2003 12–13.
- 7 Mitchell, “Roll Over Euclid” 354.
- 8 Mitchell, “Roll Over Euclid” 352.
- 9 Lenoir and Alt, “Flow, Process, Fold” 12–13.
- 10 Asli Arpak, “Physical and Virtual” (The Middle East Technical University, 2008) 44.
- 11 John Frazer, speaking at the Designing the Dynamic Conference, suggested that the adoption of CAD, and the abandonment of generative approaches was driven by several factors. First was that the lack of computing power available at the time was insufficient. Second was the homogenising effects of major CAD companies that dictated the expected ‘needs’ of architects, and forced designers away from open systems.
- 12 Tim Schork, “Option Explicit,” in *First International Conference on Critical Digital: What Matters(s)?*, 2008 41.
- 13 Mario Carpo, “Digital Style,” *Log* (2011) 41.
- 14 Carpo, “Digital Style” 41.
- 15 Mark Burry, *Scripting Cultures: Architectural Design and Programming* (Wiley, 2011) 31.
- 16 There is a strong tie between the surface strategies that define contemporary computational architecture and the surface strategies of AA-influenced landscape urbanism. The landscape and the façade both operate as surfaces for computational techniques. Both operate as blank slates upon which to place geometry that often bears little relation to the interior and exterior conditions of the building and/or site.
- 17 Robert Woodbury, *Elements of Parametric Design* (Routledge, 2010) 23–24.
- 18 Woodbury, *Elements of Parametric Design* 23–24.
- 19 Paul Coates quoted in Burry, *Scripting Cultures: Architectural Design and Programming* 81.
- 20 Mark Burry, “Paramorph,” *Architectural Design* (1999).
- 21 Robert Aish, “Compter-Aided Design Software to Augment the Creation of Form,” in *Computers in Architecture*, 1992 99.
- 22 Woodbury, *Elements of Parametric Design* 23–24.
- 23 This was first introduced in the 1960s by architects exploring cybernetics and computation, such as Christopher Alexander, John Frazer, and Cedric Price. While this view has never gone away, it is now resurgent amongst architects working with ‘systems thinking’ in areas such as sustainability and computational design.

- 24 Sean Ahlquist and Achim Menges, eds., *Computational Design Thinking* (John Wiley & Sons, 2011) 13.
- 25 Kostas Terzidis, *Expressive Form: A Conceptual Approach to Computational Design* (Spon Press, 2003).
- 26 Neal Leach quoted in Burry, *Scripting Cultures: Architectural Design and Programming* 67.
- 27 Ahlquist and Menges, *Computational Design Thinking* 13.
- 28 Burry, *Scripting Cultures: Architectural Design and Programming* 47.
- 29 Robert Somol, “Dummy Text, or the Diagrammatic Basis of Contemporary Architecture,” in *Diagram Diaries*, 1999 7.
- 30 James Corner, “Eidetic Operations and New Landscapes,” in *Recovering Landscape : Essays in Contemporary Landscape Architecture*, 1999 153.
- 31 Peter Wood, “Drawing the Line: A Working Epistemology for the Study of Architectural Drawing” (The University of Auckland, 2002) 43.
- 32 Quoted in Wood, “Drawing the Line: A Working Epistemology for the Study of Architectural Drawing” 22.
- 33 Pia Ednie-Brown, “All-Over, Over-All: biothing and Emergent Composition,” *Architectural Design* (2006) 38.
- 34 Burry, “Paramorph.”
- 35 Schork, “Option Explicit” 42–43.
- 36 Burry, “Paramorph.”
- 37 Neil Leach, “Digital Morphogenesis,” *Architectural Design* (2009) 35.
- 38 See the endless proliferation of folded slabs, ruptured plans, and complex curves.
- 39 Tom Wiscombe, “Emergent Processes” (2009).
- 40 Dimitrie Stefanescu, “Architecture and The Digital. From Modernity to Nonmodernity” (2011).
- 41 Ahlquist and Menges, *Computational Design Thinking* 181–182.
- 42 Ahlquist and Menges, *Computational Design Thinking* 20.
- 43 Stan Allen, “Diagrams Matter,” *ANY - Architecture New York* (1998) 17.
- 44 Ahlquist and Menges, *Computational Design Thinking* 181–182.
- 45 Ahlquist and Menges, *Computational Design Thinking* 8.
- 46 Casey Reas quoted in Burry, *Scripting Cultures: Architectural Design and Programming*, no. 55–58.
- 47 Karl Chu, “Metaphysics of Genetic Architecture and Computation,” *Architectural Design* (2006) 39.
- 48 Chu, “Metaphysics of Genetic Architecture and Computation” 42.
- 49 Schork, “Option Explicit” 42.
- 50 Stefanescu, “Architecture and The Digital. From Modernity to Nonmodernity.”
- 51 Werner Heisenberg, *Physics and Philosophy: The Revolution in Modern Science* (Harper Perennial Modern Classics, 2007) 32.
- 52 Aldo Leopold, *A Sand County Almanac* (Oxford University Press, 1949) 216.
- 53 Olmstead’s work, particularly Back Bay Fens being an obvious exemplar of this, along with his other work and that of Humphry Repton, Jens Jensen, and others.
- 54 Peter Connolly, “Embracing Openess: Making Landscape Urbanism Landscape Architectural: Part One,” in *The Mesh Book: Landscape/ Infrastructure*, 2004 81–82.
- 55 Susan Herrington, “The Nature of Ian McHarg’s Science,” *Landscape Journal* (2010).

56 Ian McHarg and Frederick Steiner, *The Essential Ian McHarg* (Island Press, 2007) 101.

57 C. Howett, “Ecological Values in Twentieth-Century Landscape Design: A History and Hermeneutics,” *Landscape Journal* (1998) 84.

58 Frederick Steiner, “The Ghost of Ian McHarg,” *Log* (2003) 147.

59 Steiner, “The Ghost of Ian McHarg” 147.

60 It’s no coincidence that a landscape architect owns ESRI: the largest commercial GIS company. ESRI traces its roots back to work done at Harvard’s Graduate School of Design in the early 1970s.

61 Shanti Fjord Levy, “Rooting Landscape Urbanism,” *Lunch* (2008) 122.

62 Herrington, “The Nature of Ian McHarg’s Science” 14.

63 Herrington, “The Nature of Ian McHarg’s Science” 14.

64 Leanne Muir, “Mapping Landscape Urbanism” (The University of Manitoba, 2010) 45.

65 Herrington, “The Nature of Ian McHarg’s Science” 18.

66 Ian McHarg, *The Fitness of Man’s Environment* (Smithsonian Institution Press, 1968) 212.

67 Herrington, “The Nature of Ian McHarg’s Science” 11.

68 Steiner, “The Ghost of Ian McHarg.”

69 Herrington, “The Nature of Ian McHarg’s Science” 13.

70 Herrington, “The Nature of Ian McHarg’s Science” 13.

71 James Corner, “Ecology and Landscape As Agents of Creativity,” in *Ecological Design and Planning*, 1997 83.

72 James Corner, “A Discourse on Theory II: Three Tyrannies of Contemporary Theory and the Alternative of Hermeneutics,” *Landscape Journal* (1991) 117.

73 James Corner, “The Agency of Mapping,” in *Mappings*, 1999 214.

74 Corner, “The Agency of Mapping” 214.

75 Corner, “The Agency of Mapping” 229–230.

76 Corner, “The Agency of Mapping” 230.

77 Corner, “The Agency of Mapping” 230.

78 Connolly, “Embracing Openess: Making Landscape Urbanism Landscape Architectural: Part One” 202.

79 Corner, “The Agency of Mapping” 216.

80 Fjord Levy, “Rooting Landscape Urbanism” 122.

81 Corner, “The Agency of Mapping” 252.

82 We see the same fallacy within early experiments with digital architecture. Greg Lynn’s ‘animate design’ claims to embody natural forces by mapping local flows over time. Marc Treib, *Representing Landscape Architecture* (Taylor and Francis, 2008) 21–22 Yet, this just makes these forces visible, it does little to engage with their dynamics in a productive sense.

83 Chris Reed, “Landscape Urbanism in Practice” (2010).

84 Eng Kiat, “Temporal Mutability” (2008).

85 Kiat, “Temporal Mutability.”

86 Charles Waldheim, “Landscape as Urbanism,” in *The Landscape Urbanism Reader*, 2006 39.

87 Reed, “Landscape Urbanism in Practice.”

88 Richard Weller and Martin Musiatowicz, “Landscape Urbanism: Polemics Toward an Art of Instrumentality?” in *The Mesh Book: Landscape/Infrastructure*, 2004 69.

89 Sanford Kwinter, “La Citta Nuova: Modernity and Continuity,” *New York: Zone* (1986) 106.

90 An area of land in the North of Paris, Villete was was the perfect staging ground for the competition: a site that was so relevant precisely because it was so irrelevant. Once a grand project of the Napoleonic era, the abattoir was superseded just a decade after its completion; the invention of portable refrigeration having rendered its hyper-centralised model of processing and distribution unnecessary and unneeded. Ever since, the district had been in a state of progressive decay, culminating in its almost complete abandonment by the mid 20th century. Its derelict state spoke far beyond its own particularities — the woes of Villette came to represent the woes of modernity itself.

91 Nicholas Pevzner and Stephanie Carlisle, “The Performative Ground: Rediscovering The Deep Section” (2012).

92 Arpak, “Physical and Virtual” 49.

93 Examples being OMA’s scheme for Melun Seurat, the “Programmatic Lava” project; MDRDV’s “datascapes”, and FOA’s topographic buildings such as the Yokohama International Port Terminal.

94 Muir, “Mapping Landscape Urbanism” 80.

95 Muir, “Mapping Landscape Urbanism” 79.

96 Alberto Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn’,” *Monograph* (2010).

97 Muir, “Mapping Landscape Urbanism” 131.

98 Consider these two quotes, one from Waldheim, one from Mostafavi. “Landscape Urbanism describes a disciplinary realignment currently underway in which landscape replaces architecture as the basic building block of contemporary urbanism” against “this dialogue is not limited by the traditional definition of the terms ‘building’ and ‘landscape’: it allows for the simultaneous presence of the one within the other, buildings as landscapes, landscapes as buildings.” Both quotes from Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn’.”

99 Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn’.”

100 Muir, “Mapping Landscape Urbanism” 80.

101 Pevzner and Carlisle, “The Performative Ground: Rediscovering The Deep Section.”

102 Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn’.”

103 Muir, “Mapping Landscape Urbanism” 80.

104 Kiat, “Temporal Mutability.”

105 There is a strong tie between the surface strategies that define contemporary computational architecture and the surface strategies of AA-influenced landscape urbanism. The landscape and the façade both operate as surfaces for computational techniques. Both operate as blank slates upon which to place geometry that often bears little relation to the interior and exterior conditions of the building and/or site.

106 Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn’.”

107 Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn’.”

108 Pevzner and Carlisle, “The Performative Ground: Rediscovering The Deep Section.”

109 Kiat, “Temporal Mutability.”

110 Kiat, “Temporal Mutability.”

111 Fjord Levy, “Rooting Landscape Urbanism.”



112 It should also be noted that the scope of formal interventions opens up  
 questions about the practicality of the design, if just in terms of the cost and  
 complexity of actually constructing it.Kiat, “Temporal Mutability.”

113 Kiat, “Temporal Mutability.”

114 Toscano, “Landscape Urbanism, Praxis, and the ‘Spatial Turn.’”

115 Steiner, “The Ghost of Ian McHarg” 147.

116 Muir, “Mapping Landscape Urbanism” 48.

117 Pevzner and Carlisle, “The Performative Ground: Rediscovering The Deep  
 Section.”

118 Weller and Musiatowicz, “Landscape Urbanism: Polemics Toward an Art of  
 Instrumentality?” 69.

119 Weller and Musiatowicz, “Landscape Urbanism: Polemics Toward an Art of  
 Instrumentality?” 69.

120 Steiner, “The Ghost of Ian McHarg” 149.

121 Rob Barnett, N. Popov, and J. Margaretts, “Landscape Systems Modelling,”  
 in *Proceedings of the 13th Annual Australia and New Zealand Systems Conference*,  
 2007 3.

122 Steiner, “The Ghost of Ian McHarg” 149.

123 Muir, “Mapping Landscape Urbanism” 54.

124 Kiat, “Temporal Mutability.”

125 Arguably the High Line provides an example of a very engaged form-making  
 used with landscape urbanist strategies. However it remains unclear to what  
 degree it is performative and adaptive, and how this is helped or hampered  
 by its physical forms. STOSS’s Erie Street Plaza provides another high-profile  
 example of how these might work at a small scale.

126 Herrington, “The Nature of Ian McHarg’s Science” 8.

127 Judith Stilgenbauer, “Rethinking Urban Landscapes,” in *ASLA Landscape  
 Urbanism Lecture Series*, 2012.

128 See the Downsview Park Toronto Competition, where the OMA/Bruce  
 Mau entry remains unimplemented, with one of the main reasons being that  
 the open-endedness has made the project difficult to sell and implement.

129 Stilgenbauer, “Rethinking Urban Landscapes.”

130 Tim Love, “Paper Architecture, Emerging Urbanism” (2010).

131 Love, “Paper Architecture, Emerging Urbanism.”

132 As an example, Morphosis and Steven Holl, although not explicitly tied  
 to landscape urbanism, produce work that is similar to the work of the AA  
 landscape urbanists.

133 Fjord Levy, “Rooting Landscape Urbanism.”

134 Fjord Levy, “Rooting Landscape Urbanism.”

135 Peter Connolly, “What is Design Research in Landscape Architecture?” in  
*Technique*, 2002 22–23.

136 Nadia Amoroso, *Representing Landscapes: A Visual Collection of Landscape  
 Architectural Drawings*, ed. Nadia Amoroso (Routledge, 2012) 63.

137 Connolly, “What is Design Research in Landscape Architecture?” 22–23.

138 Sue-Anne Ware, “Representation, Generation and Emancipation in Virtual  
 Landscapes,” *Landscape Review* (2003) 3.

139 Connolly, “Embracing Openess: Making Landscape Urbanism Landscape  
 Architectural: Part One” 83.

140 Connolly, “Embracing Openess: Making Landscape Urbanism Landscape  
 Architectural: Part One” 83.

141 Connolly, “Embracing Openess: Making Landscape Urbanism Landscape  
 Architectural: Part One” 84.

142 Connolly, “What is Design Research in Landscape Architecture?” 23–24.

143 Fjord Levy, “Rooting Landscape Urbanism.”

144 Peter Connolly, “What is at Hand?” *Kerb* (1999) 75.

145 Ware, “Representation, Generation and Emancipation in Virtual Landscapes”  
 6.

146 Barnett, Popov, and Margaretts, “Landscape Systems Modelling” 1.

147 Connolly, “What is at Hand?” 81.

148 Alessio Erioli, Mirco Bianchini, and Peiro Bruschi, “Interwoven Landscape,”  
 in *Proceedings of the 2011 International Conference on Green Buildings and  
 Sustainable Cities*, 2011 3.

149 Treib, *Representing Landscape Architecture* 100.

150 Treib, *Representing Landscape Architecture* 100.

151 Fjord Levy, “Rooting Landscape Urbanism.”

152 Erioli, Bianchini, and Bruschi, “Interwoven Landscape” 3.

153 Farshid Moussavi, “Parametric Software is no Substitute for Parametric  
 Thinking” (2011).

154 Christopher Peter Griffiths, “Reframing the Given” (Unitec Institute of  
 Technology, 2005) 10.

155 Woodbury, *Elements of Parametric Design* 1.

156 Arpak, “Physical and Virtual” 50.

157 Barnett, Popov, and Margaretts, “Landscape Systems Modelling” 1.

158 Eduardo Rico, Alfredo Ramirez, and Eva Castro, “Projective Readings:  
 Indexes and Diagrams in Landscape Urbanism,” in *Representing Landscapes: A  
 Visual Collection of Landscape Architectural Drawings*, 2012 22.

159 Arpak, “Physical and Virtual” 50–51.

160 Woodbury, *Elements of Parametric Design* 8.

161 Woodbury, *Elements of Parametric Design* 8.

162 Woodbury, *Elements of Parametric Design* 201.

163 Woodbury, *Elements of Parametric Design* 187.

164 Richard Helm Ralph Johnson John Vlissides Erich Gamma, *Design Patterns:  
 Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional,  
 1998) 233.

165 Erich Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*  
 233.

166 Its worth noting that as a trained mathematician, Alexander was one of them  
 few architects who had the expertise needed to program a computer for these  
 purposes.

167 Molly Wright Steenson, *Artificial Intelligence, Architectural Intelligence: The  
 Computer in Architecture, 1960–80*, 2010 4–5.

168 While the book is probably the most-widely read treatise on architecture, it  
 is also extremely unlikely to be assigned to the reading list of an architecture  
 course.William Saunders, “Review: A Pattern Language,” *Harvard Design  
 Magazine* (2002) 1–2

169 Christopher Alexander, Sara Ishikawa, and Murray Silverstein, *A Pattern  
 Language* (Oxford University Press, 1977) 656.

170 Alexander, Ishikawa, and Silverstein, *A Pattern Language* 551.

171 Carl Douglas, “Where Does Christopher Alexander Go Wrong?” (2011).

172 Douglas, “Where Does Christopher Alexander Go Wrong?”

173 Saunders, “Review: A Pattern Language” 2.

174 Not having buildings above four stories doesn't make for the most cheap,  
diverse, or sustainable city centre. The claim that "there is abundant evidence  
to show that high rise buildings make people crazy" seems to have not borne  
out, to say the least. Alexander, Ishikawa, and Silverstein, *A Pattern Language*  
115

175 Open source software being software where the code is made available, so  
that programs can be modified and redistributed.

176 This would be a particularly hard thing to do, as they particular code needed  
to a zip file is complex, and is potentially outside the programmer's area  
of expertise. Thus modules are popular for tedious or complex pieces of  
functionality.

177 Woodbury, *Elements of Parametric Design* 8.

178 Woodbury, *Elements of Parametric Design* 27–28.

179 Woodbury, *Elements of Parametric Design* 30.

180 Woodbury, *Elements of Parametric Design* 38.

181 Woodbury, *Elements of Parametric Design* 36–37.

182 Davor Babie, "Visual Programming and Why It Sucks" (n.d.).

183 David Rutten, "Round 3: A rebuttal by ayg011" (2012).

184 Python was chosen as it is a much easier language to both read and write; it  
abstracts away much unnecessary complexity.

185 Primarily the use of loops, modules, and objects that were described  
previously.

186 Francis Bitonti, "Computational Tectonics" (2010) 82.

187 Tsukasa Takenaka and Aya Okabe, "Development of the Seed Scattering  
System for Computational Landscape Design," *International Journal of*  
*Architectural Computing* (2012) 424–425.

188 The 3D textures use a "2.5D" approach where several textures of the plan  
in elevation are rotated around a radius, which creates the effect of 3D when  
viewed from any angle other than a plan view.

189 Takenaka and Okabe, "Development of the Seed Scattering System for  
Computational Landscape Design" 425.

190 All of patterns had only been made available to the public in the days before  
this thesis was submitted. One pattern had been available for download a  
few months prior, and had netted approximately 1000 downloads over the  
course of several months. In their now-published form, the code is easily  
downloaded, with user-modifications and improvements able to be submitted  
back into the master copy of the code.

191 "Vision Seaview Gracefield 2030," *Hutt City Council* (2010) 32–33.

192 "Vision Seaview Gracefield 2030" 32–33.

193 "Vision Seaview Gracefield 2030" 32–33.

194 "Vision Seaview Gracefield 2030" 31.

195 Leigh Stevens and Barry Robertson, *Waiwhetu Stream 2012*, 2012 12.

196 Stevens and Robertson, *Waiwhetu Stream 2012* 12.

197 "Plants" here meant focusing on native species and significant clusters of  
plants. Scrub, grassland, and sparsely populated areas were ignored.

198 Amoroso, *Representing Landscapes: A Visual Collection of Landscape Architectural*  
*Drawings* 81.

199 Mitchell, "Roll Over Euclid" 357.

200 Ware, "Representation, Generation and Emancipation in Virtual  
Landscapes."

201 Hiroshi Ishii, Assaf Biderman, and Carlo Ratti, "Continuous Tangible  
Interfaces," *BT Technology Journal* (2004).

202 Moussavi, "Parametric Software is no Substitute for Parametric Thinking."

203 Simone Brott, "Deleuze and Intercessors," *Log* (2010) 135.

204 Andrew Ballantyne, *Deleuze and Guattari for Architects* (Routledge, 2007)  
102–103.

205 Douglas Spencer, "Architectural Deleuzism," *Radical Philosophy* (2011) 9–11.

206 Mark Wigley, "Conceptual Caretakers," in *Territorial Investigations*, 1999 176.

207 Deleuze's work contains a meta-interest in the mediums of text and  
language. One of the most notable examples of this is that each of his major  
books espouses a consistent set of ideas that are expressed with different  
terminologies and lines of argument. Such linguistic games may aid in  
understanding, but create problems when referencing his ideas. Within  
this thesis I use the lexicon established by Manuel De Landa in **Intensive  
Science and Virtual Philosophy**, whereby particular concepts are referred  
to by a semi-standardised series of names. DeLanda's system was chosen for  
his emphasis on the analytic and scientific portions of Deleuze's analysis, and  
because of his favoured position as an expert in the applications of Deleuzian  
theory within software, art, and architecture.

208 Ontologies articulate the most fundamental assumptions that define reality.  
An ontological system attempts to define the manner in which entities can  
be said to be exist. Primarily, it is that which causes certain phenomena to be  
identified as discrete objects, and how such discrete objects can be correlated  
to, or distinguished from, one another.

209 "The Philosophy of Gilles Deleuze," in , 2006.

210 Kant, Hume, etc.

211 Plato etc.

212 "The Philosophy of Gilles Deleuze."

213 "The Philosophy of Gilles Deleuze."

214 John Rajchman, *The Deleuze Connections* (MIT Press, 2000) 23.

215 Ballantyne, *Deleuze and Guattari for Architects* 11.

216 An extremely reductive simplification, ignoring the roles of behaviour,  
memory, interactionm etc.

217 Temperature, pressure, velocity and viscosity are also examples of intensive  
properties.

218 Mass, length, energy, electrical resistance and many other properties are also  
extensive.

219 This is actually a simplification, but a necessary one. Topological spaces can  
only been seen as curved in relation to being incarnated within Euclidian  
space. Topology exists prior to metric curvatures, yet this remains the only  
way in which to effectively visualise it.

220 Richard Bishop and Richard Crittenden, *Geometry of Manifolds* (Academic  
Press, 1964) 39.

221 Bishop and Crittenden, *Geometry of Manifolds* 39.

222 Manuel De Landa, "Deleuze and the Genesis of Form," *Art Orbit* (1998).

223 Manuel De Landa, *Intensive Science & Virtual Philosophy* (Continuum, 2005)  
69.

224 Ronquillo Ulysses, "On Delanda's Intensive Science and Virtual Philosophy"  
(2008).

225 Sanford Kwinter, "Who's Afraid of Formalism?" in *Phylogenesi: FOA's Arc*,  
2004 96.

226 Manuel De Landa and Idhe Don, “1000 Years of War: CTHEORY  
Interview with Manuel De Landa” (2003).

227 Janet Bayly, *The Heretaunga / Waiwhetu River Mouth* (Petone Settlers Museum,  
1988) 12.

228 Bayly, *The Heretaunga / Waiwhetu River Mouth* 12.

229 “History of the Hutt Valley” (n.d.).

230 Bayly, *The Heretaunga / Waiwhetu River Mouth* 10.

231 “History of the Hutt Valley.”

232 Bayly, *The Heretaunga / Waiwhetu River Mouth* 15.

233 Bayly, *The Heretaunga / Waiwhetu River Mouth* 17.

234 Bayly, *The Heretaunga / Waiwhetu River Mouth* 19.

235 “History of the Hutt Valley.”

236 Bayly, *The Heretaunga / Waiwhetu River Mouth* 32.

237 “History of the Hutt Valley.”

238 “History of the Hutt Valley.”

239 Bayly, *The Heretaunga / Waiwhetu River Mouth* 45.

240 Bayly, *The Heretaunga / Waiwhetu River Mouth* 46.

241 C. Treadwell, *The Hutt River* (The Hutt River Board, 1959) 25.

242 Treadwell, *The Hutt River* 29.

243 Treadwell, *The Hutt River* 30.

244 Treadwell, *The Hutt River* 34–35.

245 Treadwell, *The Hutt River* 34–35.



## Bibliography

Addington, Michelle. “Contingent Behaviours.” *Architectural Design* (March 28, 2009): 1–6.

Ahlquist, Sean, and Achim Menges, eds. *Computational Design Thinking*. John Wiley & Sons, 2011.

Aish, Robert. “Compter-Aided Design Software to Augment the Creation of Form.” In *Computers in Architecture*, 97–104. Harlow, UK: Longman, 1992.

Alexander, Christopher. *The Timeless Way of Building*. Oxford University Press, 1979.

Alexander, Christopher, and Peter Eisenman. “Contrasting Concepts of Harmony in Architecture.” Edited by Lucien Steil. *Katarxis*, March 6, 2004. [http://www.katarxis3.com/Alexander\\_Eisenman\\_Debate.htm](http://www.katarxis3.com/Alexander_Eisenman_Debate.htm).

Alexander, Christopher, Sara Ishikawa, and Murray Silverstein. *A Pattern Language*. Oxford University Press, 1977.

Allen, Stan. “Diagrams Matter.” *ANY - Architecture New York* 23 (1998): 16–19.

Allen, Stan. “From Object to Field.” *Architectural Design* 67, no. 5 (1997): 24–31.

Amoroso, Nadia. *Representing Landscapes: a Visual Collection of Landscape Architectural Drawings*. Edited by Nadia Amoroso. 1st ed. Routledge, 2012.

Andrei Stefanescu, Dimitrie. “Relinquishing Control: Reactions to Emergence.” *Improved.Ro*, March 27, 2011. <http://improved.ro/blog/?p=1014>.

Anne Marie, Fourcade. “Architecture and Automatized Methods: Criticisms on the Current Issues.” Massachusetts Institute of Technology, 1975.

Anson Coons, Steven. “An Outline of the Requirements for a Computer-Aided Design System” (January 7, 2010): 1–6.

Arpak, Asli. “Physical and Virtual.” The Middle East Technical University, 2008.

Babie, Davor. “Visual Programming and Why It Sucks.” *Dado*, n.d. <http://blog.davor.se/blog/2012/09/09/Visual-programming>.

Ballantyne, Andrew. *Deleuze and Guattari for Architects*. Routledge, 2007.

Barnett, Rob. *A Ten Point Guide to Urban Field Theory*, September 8, 2010.

Barnett, Rob, N Popov, and J Margaretts. “Landscape Systems Modelling.” In *Proceedings of the 13th Annual Australia and New Zealand Systems Conference*, edited by J Shepherd and K Fielder, 2007.

Bauer, Kirsten, Chris Sawyer, and Peter Connolly. “Against the Picturesque, What Specificity Offers Us as Designers.” *Kerb* (March 20, 2012): 1–4.

Bayly, Janet. *The Heretaunga / Waiwhetu River Mouth*. Petone Settlers Museum, 1988.

Beigel, Florian, and Philip Christou. “Landscapes of the Second Nature.” *Architectural Design* (February 17, 2007): 1–10.

Bessa, Maria. “Algorithmic Design.” *Architectural Design* 79, no. 1 (2009): 120–123.

Bishop, Richard, and Richard Crittenden. *Geometry of Manifolds*. New York: Academic Press, 1964.

Bitonti, Francis. “Computational Tectonics.” *Acadia 09*. Chicago, January 3, 2010. [http://ar.fadarch.com/FADarch\\_acadia09.pdf](http://ar.fadarch.com/FADarch_acadia09.pdf).

Boden, Margaret, and Ernest Edmonds. “What Is Generative Art?.” *Digital Creativity* 20 (March 2009): 21–46.

Brott, Simone. “Deleuze and Intercessors.” *Log* 18 (April 15, 2010): 135–151.

Bryant, Levi. “Problems of Immanence.” *Larval Subjects*, May 12, 2009. <http://larvalsubjects.wordpress.com/2009/05/12/problems-of-immanence/>.

Burry, Mark. “Digital Applications and Cultural Heritage: a Dynamic Co-Evolution.” In *The Proceedings of DACH 2007*, edited by Liam C, Y Liu, and J Hou, 417–442. Taipei, 2007.

Burry, Mark. “Paramorph.” *Architectural Design*, no. 2 (March 4, 1999): 78–83.

Burry, Mark. *Scripting Cultures: Architectural Design and Programming*. 1st ed. Wiley, 2011.

Burry, Mark, and Andrew Maher. “The Parametric Bridge.” In *Acadia 22*, 23–26. Indianapolis, 2003.

Busby, Peter, Max Richter, and Michael Driedger. “Towards a New Relationship with Nature.” *Architectural Design* (October 11, 2011): 1–8.

Cache, Bernard. *A Plea for Euclid*. ANY (Architecture New York), 1999.

Carenini, Cristian. “The Diagrammatic Thought in Landscape Urbanism,” June 9, 2009. [http://gerhard\\_dirmoser.public1.linz.at/FU/ARCH\\_Literatur\\_Liste.pdf](http://gerhard_dirmoser.public1.linz.at/FU/ARCH_Literatur_Liste.pdf)

Carmo, Mario. “Digital Style.” *Log* (August 17, 2011): 41–52.

Ceccato, Cristiano. “Evolutionary Design Tools for Mass-Customisation” (March 12, 2012): 1–7.

Chang, Liam. “The Coyote Is Always More Humiliated Than Harmed by His Failures..” *Archinect*, January 26, 2011. <http://archinect.com/lian/the-coyote-is-always-more-humiliated-than-harmed-by-his-failures>.

Chang, Wei-Tsang, and Teng-Wen Chang. “Folding Space with Time-Based Operations.” In *Proceedings of the 11th International Conference on Computer Aided Architectural Design Research in Asia*, 569–571, Kumamoto, 2006.

Chu, Karl. “Metaphysics of Genetic Architecture and Computation.” *Architectural Design* 76, no. 4 (2006): 38–45.

Chu, Karl. *The Turing Dimension*, October 9, 2009.

Clarke, Joseph. “Into a Forest of Script.” *Log* (December 9, 2009): 1–9.

Coates, Paul S. *Programming Architecture*. Taylor & Francis, 2010.

Committee, City Development. *Vision Seaview Gracefield 2030 Implementation Strategy and Work Plan*. Hutt City Council, February 18, 2011.

Connolly, Peter. “Embracing Openess: Making Landscape Urbanism Landscape Architectural: Part One.” In *The Mesh Book: Landscape/Infrastructure*, edited by Julian Raxworthy and Jessica Blood, 76–104. Melbourne: RMIT University Press, 2004.

Connolly, Peter. “Embracing Openess: Making Landscape Urbanism Landscape Architectural: Part Two.” In *The Mesh Book: Landscape/Infrastructure*, edited by Julian Raxworthy and Jessica Blood, 200–220. Melbourne: RMIT University Press, 2004.

Connolly, Peter. “What Is at Hand?.” *Kerb* (November 9, 2011): 1–14.

Connolly, Peter. “What Is Design Research in Landscape Architecture?.” In *Technique*. RMIT University, 2002.

Coombes, Daniel. “Unfamiliar Terrain.” Unitec Institute of Technology, 2011.

Corner, James. “A Discourse on Theory I: ‘Sounding the Depths’—Origins, Theory, and Representation.” *Landscape Journal* 9 (1990): 61–78.

Corner, James. “A Discourse on Theory II: Three Tyrannies of Contemporary Theory and the Alternative of Hermeneutics.” *Landscape Journal* 10, no. 2 (1991): 115–133.

Corner, James. “Ecology and Landscape as Agents of Creativity.” In *Ecological Design and Planning*, edited by George Thompson and Frederick Steiner, 81–107. Wiley-Academy, 1997.

Corner, James. “Eidetic Operations and New Landscapes.” In *Recovering Landscape : Essays in Contemporary Landscape Architecture*, edited by James Corner, 153–169. New York: Princeton Architectural Press, 1999.

Corner, James. “The Agency of Mapping.” In *Mappings*, edited by Denis Cosgrove, 213–253. London: Reaktion Books, 1999.

Council, Greater Wellington Regional. “Waiwhetu Stream Clean-Up and Flood Protection Works,” July 9, 2009.

Cox, Geoff. “Generator: About Generative Art and/or Software Art.” *Anti-Thesis*, June 9, 2006. <http://www.anti-thesis.net/contents/texts/generator/generator.pdf>.

Danahy, John. “Advanced Computation in Landscape Architecture.” *University of Toronto*, July 17, 2006. [http://www.daniels.utoronto.ca/courses/200607/lan\\_3025h](http://www.daniels.utoronto.ca/courses/200607/lan_3025h).

Davis, Daniel. “Evolving Digital Morphogenesis.” Victoria University of Wellington, 2009.

Davis, Daniel. “The Architect and Neil Leach’s Digital Morphogenesis.” Victoria University of Wellington, 2009.

Davis, Daniel. “Untangling Grasshopper – Part 1: Design Patterns.” *Digital Morphogenesis*, n.d. <http://www.nzarchitecture.com/blog/index.php/2010/07/31/untangling-grasshopper-part-1-design-patterns/>.

Davis, Daniel. “Untangling Grasshopper – Part 2: Optimisation.” *Digital Morphogenesis*, n.d. <http://www.nzarchitecture.com/blog/index.php/2010/08/11/untangling-grasshopper-part-2-optimisation/>.

Davis, Daniel, Jane Burry, and Mark Burry. “Untangling Parametric Schemata : Enhancing Collaboration Through Modular Programming.” In *Proceedings of CAAD Futures 2011*, 55–78, 2011.

Davis, Daniel, Mark Burry, and Jane Burry. “The Flexibility of Logic Programming.” In *Proceedings of the 16th International Conference on Computer Aided Architectural Design Research in Asia* 29–38, Newcastle, 2011.

Davis, Daniel, Flora Salim, and Jane Burry. “Designing Responsive Architecture.” In *Circuit Bending, Breaking and Mending: Proceedings of the 16th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2011*, edited by C Herr, S Roudavsky, and A Schnabel, 1–10, 2011.

De Landa, Manuel, and Idhe Don. “1000 Years of War: CTHEORY Interview with Manuel De Landa.” *Ctheory*, January 5, 2003. <http://ctheory.net/articles.aspx?id=383>.

De Landa, Manuel. “Deleuze and the Genesis of Form.” *Art Orbit* 1 (March 1998).

De Landa, Manuel. “Immanence and Transcendence in the Genesis of Form.” *A Deleuzian Century?* 96, no. 3 (1997): 499–514.

De Landa, Manuel. *Intensive Science & Virtual Philosophy*. Continuum, 2005.

De Landa, Manuel. *Philosophies of Design: the Case of Modeling Software*. Verb: Processing, 2001.

De Landa, Manuel. *Philosophy & Simulation: the Emergence of Synthetic Reason*. Continuum, 2011.

De Landa, Manuel. “The Philosophy of Gilles Deleuze.” European Graduate School, Saas-Fee, Switzerland, 2006.

De Landa, Manuel, and Neil Leach. “The Limits of Urban Simulation.” *Architectural Design* (June 16, 2009): 1–6.

De Wolf, Tom, and Tom Holvoet. “Emergence Versus Self-Organisation: Different Concepts but Promising When Combined.” *Citeseerx*, March 6, 2005. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.2913&rep=rep1&type=pdf>.

Deleuze, Gilles. *Difference and Repetition*. New Ed. Continuum International Publishing Group Ltd., 1968.

Deleuze, Gilles, and Felix Guattari. *A Thousand Plateaus: Capitalism and Schizophrenia*. University of Minnesota Press, 1987.

Diaz Alonso, Herman, and Thom Mayne. “Generation(S) and the Generative: a Conversation.” *Log* 17 (June 6, 2009): 1–5.

Digital Landscape Architecture Now, Thames & Hudson Ltd, 2012.

Douglas, Carl. “Where Does Christopher Alexander Go Wrong?.” *Diffusive. Wordpress.com*, December 9, 2011. <http://diffusive.wordpress.com/2011/12/09/where-does-christopher-alexander-go-wrong/>.

Ednie-Brown, Pia. “All-Over, Over-All: Biothing and Emergent Composition.” *Architectural Design* 76, no. 4 (July 27, 2006): 72–81.

Ednie-Brown, Pia. “Plastic Super Models: Aesthetics, Architecture and the Model of Emergence.” *Fibreculture Journal*, n.d. [http://www.journal.fibreculture.org/issue12/issue12\\_ednie-brown\\_print.html](http://www.journal.fibreculture.org/issue12/issue12_ednie-brown_print.html).

Ednie-Brown, Pia. “The Aesthetics of Emergence.” RMIT University, 2007.

Ednie-Brown, Pia. “The Texture of Diagrams.” *Diagrammania*, no. 74 (June 26, 2000): 1–10.

Ednie-Brown, Pia. “The Will to Animation.” *Architectural Design* 71, no. 2 (June 27, 2001): 64–73.

Enns, Jonothan. “Intelligent Wood Assemblies.” *Architectural Design* (October 23, 2010): 1–6.

Erich Gamma, Richard Helm Ralph Johnson John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1998.

Erioli, Alessio, Mirco Bianchini, and Peiro Bruschi. “Interwoven Landscape.” In *Proceedings of the 2011 International Conference on Green Buildings and Sustainable Cities*, 1–9, 2011.

Evans, Robin. *Translations From Drawing to Building and Other Essays*. AA Publications, 2005.

Feast, Luke. “The Discrete and the Continuous in Architecture and Design.” In *Proceedings of Wonderground, the 2006 International Conference of the Design Research Society*, edited by Ken Friedman, Terence Love, Eduardo Corte-Real, and Chris Rust, 1–14, Libson, 2006.

Feast, Luke. “The Science of Multiplicities: Post-Structuralism and Ecological Complexities in Design,” Victoria University of Wellington, 2006.

Fernando, Ruwan, Robin Drogemuller, Jane Burry, and Flora Salim. “Patterns, Heuristics, for Architectural Design Support” (March 16, 2010): 1–10.

Fischer, T, Mark Burry, and Robert Woodbury. “Object-Oriented Modelling Using XML in Computer-Aided Architectural and Education CAD.” In *Proceedings of the Fifth Conference on Computer-Aided Architectural Design Research in Asia*, 145–155, Singapore, 2000.



Fjord Levy, Shanti. “Rooting Landscape Urbanism.” *Lunch* 3 (April 7, 2008): 113–130.

Foster Gage, Mark. “Architecture in the Age of Dissensus.” *Fulcrum* (November 21, 2011): 1–2.

Foster, Hal. “Post-Critical.” *October* (March 2, 2012): 1–6.

Frazer, John. *An Evolutionary Architecture*. Architectural Association Publications, 1995.

Frazer, John. “Intentionality: the Coding of a Design Concept.” The Architecture Association, November 16, 2011. <http://www.aaschool.ac.uk/VIDEO/lecture.php?ID=1633>.

Frazer, John. “The Architectural Relevance of Cybernetics.” *Systems Research* 10, no. 3 (1993): 43–48.

Gairbheith, Karol. “Parametric Design: Towards a Future Architecture,” London South Bank University, 2009.

Gillespie, Tarleton. “The Relevance of Algorithms.” Edited by Tarleton Gillespie and Pablo Boczkowski. *Media Technologies* (November 26, 2012): 1–32.

Group, Waiwhetu Stream Working Group. *Waiwhetu Stream Action Plan*, November 2, 2000.

Heisenberg, Werner. *Physics and Philosophy: the Revolution in Modern Science*. Harper Perennial Modern Classics, 2007.

Hensel, Michael. “Computing Self-Organisation: Environmentally Sensitive Growth Modelling.” *Architectural Design* 76, no. 2 (2006): 12–17.

Hensel, Michael, and Achim Menges. “Patterns in Performance-Oriented Design.” *Architectural Design* (October 29, 2009): 1–6.

Herrington, Susan. “The Nature of Ian McHarg’s Science.” *Landscape Journal* 29, no. 1 (2010): 1–20.

Herzogenrath, Bernd, ed. *Deleuze/Guattari & Ecology*. Palgrave MacMillan, 2009. <http://>.

“History of the Hutt Valley.” *Huttvalleynz.com*, n.d. <http://www.huttvalleynz.com>.

Hohmann, Heidi, and Joern Langhorst. “Landscape Architecture: an Apocalyptic Manifesto.” *Iowa State University, Department of Landscape Architecture*, October 20, 2004. <http://www.public.iastate.edu/~isitdead/printpage.html>.

Holm, Lorens, and Paul Guzzardo. “Is There a Digital Future Landscape Terrain?.” *Architectural Design* (February 17, 2007): 1–3.

Howett, C. “Ecological Values in Twentieth-Century Landscape Design: a History and Hermeneutics.” *Landscape Journal* 17, no. 2 (1998): 80–98.

Illner, Peer. “The Origins of Theory.” Edited by Jack Self and Graham Baldwin. *Fulcrum*, no. 61 (January 7, 2013): 1–2.

Ishii, Hiroshi, Assaf Biderman, and Carlo Ratti. “Continuous Tangible Interfaces.” *BT Technology Journal* 22, no. 4 (2004): 287–299.

Jaeschke, Aleksandra. “Environmental Intesifiers.” *Architectural Design* (February 14, 2008): 1–8.

Jones, Wes. “Big Forking Dilemma.” *Harvard Design Magazine* 32 (July 2, 2010): 1–10.

Kaspori, Dennis. “A Communism of Ideas.” *Rixc.Lv*, March 26, 2011. <http://www.rixc.lv/ram/en/public02.html>.

Kehne, Holger, and Jeff Turko. “Parallel Indexing.” *Architectural Design* (June 16, 2009): 1–4.

Keller, Sean. “Fenland Tech.” *Grey Room* 23 (April 12, 2006): 40–65.

Kerkidou, Maria. “Force Fields and Evolution in Design Space.” Bartlett School of Graduate Studies, 2006.

Kerslake, Christian. “Deleuze and the Meanings of Immanence.” *After 68* (June 16, 2009): 1–36.

Kerslake, Christian. *Immanence and the Vertigo of Philosophy*, 2009.

Keulemans, Guy. “Strategies for Generative Designers and the Development and Use of Generative Software Tools,” The University of New South Wales, 2006.

Khabazi, Zubin. *Generative Algorithms Using Grasshopper*, London: Architectural Association, 2010.

Kiat, Eng. “Temporal Mutability.” *Random Rants*, May 3, 2008. <http://engkiat.wordpress.com/2008/05/03/temporal-mutability-post-structuralism-and-the-indeterminate-in-the-discourse-of-landscape-urbanism/>.

Kilian, Alex. “Design Exploration Through Bidirectional Modeling of Constraints,” Massachusetts Institute of Technology, 2006.

Kirkegaard, P H, M Hougaard, and J W Stærdahl. *On Computational Fluid Dynamics Tools in Architectural Design*, Aalborg University, January 8, 2009.

Koutamanis, Alexander. “A Biased History of CAAD.” Delft University of Technology, 2005.

Kwinter, Sanford. *Architectures of Time*. 1st ed. Cambridge Mass.: MIT Press, 2002.

Kwinter, Sanford. “La Città Nuova: Modernity and Continuity.” *New York: Zone* ½ (1986).

Kwinter, Sanford. “Landscapes of Change: Boccioni’s “Stati D’animo” as a General Theory of Models.” *Assemblage* 19 (April 4, 1992): 50–65.

Kwinter, Sanford. “The Computational Fallacy.” In *Computational Design Thinking*, edited by Achim Menges and Seab Ahlquist, 211–215. Wiley, 2011.

Kwinter, Sanford. “Who’s Afraid of Formalism?.” In *Phylogensis: FOA’s Arc*. Actar, 2004.

Latour, Bruno, and Albena Yaneva. “Give Me a Gun and I Will Make All Buildings Move.” In *Explorations in Architecture: Teaching, Design, Research*, edited by R. Geiser, 80–89. An Ant’s View of Architecture. Basel: Birkhäuser, 2008.

Leach, Neil. “Digital Morphogenesis.” *Architectural Design* 79, no. 1 (2009): 32–37.

Leach, Neil, and Patrik Schumacher. “On Parametricism.” *Time + Architecture*, no. 5 (n.d.).

Lenoir, Timothy, and Casey Alt. “Flow, Process, Fold.” In *Science, Metaphor, and Architecture*, edited by Antoine Picon and Alessandra Ponte, Princeton University Press, 2003.

Leopold, Aldo. *A Sand County Almanac*. Oxford University Press, 1949.

Lopes, Dominic. *A Philosophy of Computer Art*. New York: Routledge, 2010.

Lopes, Domonic. “The Ontology of Interactive Art.” *Journal of Aesthetic Education* 35, no. 4 (March 6, 2001): 65–81.

Love, Tim. “Paper Architecture, Emerging Urbanism.” *The Design Observer*, April 13, 2010. <http://places.designobserver.com/feature/paper-architecture-emerging-urbanism/12684/>.

Management, Wriggle Coastal. “Hutt Estuary Fine Scale Monitoring 2010/11” (June 14, 2011): 1–29.

Mark, Burry, Joeph Serrano, and Juan Melero. “The Need to Step Beyond Conventional Architectural Software.” In *Education in Computer Aided Architectural Design: 11th eCAADe Conference Proceedings*, 5–25 Eindhoven, 1993.

Marsh, Andrew, and Azam Khan. “Simulation and the Future of Design Tools for Ecological Research.” *Architectural Design* (October 11, 2011): 1–10.

Massumi, Brian. “Sensing the Virtual, Building the Insensible.” *Architectural Design* 68, no. 5 (June 1, 1998): 16–24.

Matthews, Stanley. “The Fun Palace as Virtual Architecture.” *Journal of Architectural Education* (January 8, 2006): 1–10.

McCormack, Jon, Alan Dorin, and Troy Innocent. “Generative Design: a Paradigm for Design Research.” In *Proceedings of Futureground, Design Research Society*. Melbourne, 2004.

McCullough, Malcolm. “20 Years of Scripted Space.” *Architectural Design* 76, no. 4 (2006): 12–15.

McGarth, Brian, and Victoria Marshall. “Operationalising Patch Dynamics.” *Architectural Design* (February 17, 2007): 1–8.

McHarg, Ian. *The Fitness of Man’s Environment*. Washington DC: Smithsonian Institution Press, 1968.

McHarg, Ian, and Frederick Steiner. *The Essential Ian McHarg*. Island Press, 2007.

Menges, Achim, and Sean Ahlquist. “Physical Drivers: Synthesis of Evolutionary Developments and Force-Driven Design.” *Architectural Design* (January 24, 2012): 1–8.

Mennan, Zeynep. “The Question of Non Standard Form.” *Metu JFA* 25, no. 2 (January 6, 2008): 171–183.

Mitchell, William. “Roll Over Euclid.” In *Frank Gehry, Architect*, edited by Fiona Ragheb, 352–363. How Frank Gehry Designs and Builds. New York: Solomon R. Guggenheim Foundation, 2003.

Moloney, Jules. “Collapsing the Tetrahedron: Architecture with(in) Digital Machines.” In *Computers and History of Art Conference Proceedings*, Vol. 2, 2000. <http://www.chart.ac.uk/chart1999/papers/noframes/moloney.html>.

Moloney, Jules. *Designing Kinetics for Architectural Facades*. Routledge, 2011.

Moloney, Jules. “Literal and Notional Force Fields in Architecture.” *The Journal of Architecture* 16, no. 2 (April 2011): 213–229.

Moloney, Jules. *Notation and the Architectural Event*, November 24, 2007.

More, Gregory. “Animated Techniques: Time and the Technological Acquiescence of Animation.” *Architectural Design* 71, no. 2 (2001): 20–27.

Mostafavi, Mohsen, and Ciro Najle. *Landscape Urbanism: a Manual for the Machinic Landscape*. Architectural Association Publications, 2004.

Moussavi, Farshid. “Parametric Software Is No Substitute for Parametric Thinking.” *The Architectural Review*, March 12, 2011. <http://www.architectural-review.com/issues/october-2011/parametric-software-is-no-substitute-for-parametric-thinking/8620000.article>.

Muir, Leanne. “Mapping Landscape Urbanism.” The University of Manitoba, 2010.

Murphy, S. *Waiwhetu Stream Management Plan*. Wellington Regional Council, November 17, 1996.

Nordenson, Guy, Catherine Seavitt, Adam Yarinsky, and N.Y.) Museum of Modern Art New York. *On the Water*. Museum of Modern Art, 2010.

Oxman, Rivka. “Digital Architecture as a Challenge for Design Pedagogy: Theory, Knowledge, Models and Medium.” *Design Studies* 29, no. 2 (March 2008): 99–120.

Peter Griffiths, Christopher. “Reframing the Given.” Unitec Institute of Technology, 2005.

Pevzner, Nicholas, and Stephanie Carlisle. “The Performative Ground: Rediscovering the Deep Section.” *Landscape Urbanism*, August 1, 2012. <http://landscapeurbanism.com/article/the-performative-ground/>.

Rajchman, John. *The Deleuze Connections*. Cambridge Mass.: MIT Press, 2000.

Ramos, Vitorino, and Ajith Abraham. “Swarms on Continuous Data.” In *Evolutionary Computation, 2003. CEC’03. The 2003 Congress on*, vol 2 1370–1375, 2003.

Reed, Chris. “Landscape Urbanism in Practice.” *Youth Landscape Architecture*, December 1, 2010. <http://www.youthla.org/2010/12/stoss-landscape-urbanism/>.

Regnier, Ben. “Design: a Discussion About Tools,” December 18, 2010. [http://issuu.com/bregnier/docs/nf\\_final\\_project/1](http://issuu.com/bregnier/docs/nf_final_project/1).

Reiser, Jesse. *Atlas of Novel Tectonics*. 1st ed. Princeton Architectural Press, 2006.

Rico, Eduardo, Alfredo Ramirez, and Eva Castro. “Projective Readings: Indexes and Diagrams in Landscape Urbanism.” In *Representing Landscapes: a Visual Collection of Landscape Architectural Drawings*, edited by Nadia Amoroso, 22–33. Routledge, 2012.

Roberts, Nick. “A Projective Site.” Victoria University of Wellington, 2010.

Rocker, Ingeborg. “Calculus-Based Form: an Interview with Greg Lynn.” *Architectural Design* 76, no. 4 (2006): 88–95.

Rocker, Ingeborg. “When Code Matters.” *Architectural Design* (July 27, 2006): 1–10.

Rutten, David. “Round 3: a Rebuttal by Ayg011.” *I Eat Bugs for Breakfast*, February 12, 2012. <http://ieatbugsforbreakfast.wordpress.com/2012/09/12/round-3-a-rebuttal-by-ayg011/>.

Saunders, William. “Review: a Pattern Language.” *Harvard Design Managzine* 16 (October 5, 2002): 1–7.

Schork, Tim. “Modes of Composition – a Computational Approach to Design.” *Mesne.Net*, October 16, 2009. [http://mesne.net/blog/wp-content/uploads/2010/05/Schork-Modes-of-Composition\\_A-Computational-Approach.pdf](http://mesne.net/blog/wp-content/uploads/2010/05/Schork-Modes-of-Composition_A-Computational-Approach.pdf).

Schork, Tim. “Option Explicit.” In *First International Conference on Critical Digital: What Matters(s)?*, edited by Kostas Terzidis, 41–46. Cambridge, 2008.

Schumacher, Patrik. “Convergence Versus Fragmentation.” *Fulcrum* (November 21, 2011): 1–2.

Schumacher, Patrik. “Parametric Patterns.” *Architectural Design* (October 29, 2009): 1–14.

Schumacher, Patrik. “Parametricism.” *Architectural Design* (June 16, 2009): 1–10.

Schumacher, Patrik. *The Autopoiesis of Architecture*. Wiley, 2011.

Senagala, Mahesh. “Post-Spatial Architectures: the Emergence of Time-Like Parametric Worlds.” In *Proceedings of the 2002 Annual Conference of the Association for Computer Aided Design in Architecture*, 42–49, 2002.

Shane, Grahame. “The Emergence of ‘Landscape Urbanism’.” *Harvard Design Magazine* 19 (September 4, 2003): 1–8.

Sheil, Bob. “Distinguishing Between the Drawn and the Made.” *Architectural Design* (January 24, 2012): 1–6.

Shiffman, Daniel. *The Nature of Code*, 2012.

Siddique, M, Qazi Mowla, and Mohammad Masum. “Virtuality in Architecture.” In *Proceedings of the 10th International Conference on Computer Aided Architectural Design Research in Asia*, 1:342–350, 2005.

Silver, Mike. “Towards a Programming Culture in the Design Art.” *Architectural Design* (July 27, 2006): 1–7.



Skylar, Tibbits, Arthur van der Harten, and Steve Baer. *Python for Rhinoceros 5*, 2011.

Somol, Robert. “Dummy Text, or the Diagrammatic Basis of Contemporary Architecture.” In *Diagram Diaries*, 6–25. Universe, 1999.

Spencer, Douglas. “Architectural Deleuzism.” *Radical Philosophy*, no. 168 (June 22, 2011): 1–13.

Spens, Michael. “Site/Non-Site.” *Architectural Design* (February 17, 2007): 1–6.

Spiller, Neil. “Digital Solipsism and the Paradox of the Great ‘Forgetting’.” *Architectural Design* (June 25, 2010): 1–5.

Spuybroek, Lars. “Motor Geometry.” *Arch+* 138 (October 1, 1997): 1–9.

Stefanescu, Dimitrie. “Architecture and the Digital. From Modernity to Nonmodernity.” *Improved.Ro*, December 6, 2011.

Steiner, Frederick. “The Ghost of Ian McHarg.” *Log* 13 (2003): 147–151.

Stevens, Leigh, and Barry Robertson. *Waiwhetu Stream 2009*. Management, Wriggle Coastal, June 8, 2009.

Stevens, Leigh, and Barry Robertson. *Waiwhetu Stream 2012*. Management, Wriggle Coastal, July 30, 2012.

Stilgenbauer, Judith. “Rethinking Urban Landscapes.” *ASLA Landscape Urbanism Lecture Series*. UC Berkley, January 25, 2012. <http://www.youtube.com/watch?v=0bztGdaOZ2I>.

Swaffield, Simon. “Tracing Change.” *Architectural Design* (October 29, 2009): 1–6.

Takenaka, Tsukasa, and Aya Okabe. “Development of the Seed Scattering System for Computational Landscape Design.” *International Journal of Architectural Computing* 9, no. 4 (February 8, 2012): 1–17.

Terzidis, Kostas. *Algorithmic Architecture*. Routledge, 2006.

Terzidis, Kostas. *Expressive Form: a Conceptual Approach to Computational Design*. London: Spon Press, 2003.

Theodropolou, Athina. “Architectural Authorship in Generative Design,” The Bartlett School of Graduate Studies, 2007.

Toscano, Alberto. “Landscape Urbanism, Praxis, and the ‘Spatial Turn.’” *Monograph*, no. 2 (April 22, 2010): 1–15.

Treadwell, C. *The Hutt River*. The Hutt River Board, 1959.

Treib, Marc. *Representing Landscape Architecture*. Taylor and Francis, 2008.

Tremblay, Louis, Stephen Moore, Mike van den Heuvel, and David West. “Methods for Determining the Effects of Pollution on Fishes” (March 8, 2005): 1–74.

Trummer, Peter. “Morphogenetic Urbanism.” *Architectural Design* 79, no. 4 (June 16, 2009): 64–67.

Ulysses, Ronquillo. “On Delanda’s Intensive Science and Virtual Philosophy.” *Media Assemblages*, November 24, 2008. <http://mediaecologiesresonate.wordpress.com/2008/11/24/on-delandas-intensive-science-and-virtual-philosophy/>.

Vidler, Anthony. “Diagrams of Diagrams: Architectural Abstraction and Modern Representation.” *Representations* 72 (November 6, 2007): 1–21.

Vidler, Anthony. “Whatever Happened to Ecology.” *Architectural Design* (October 23, 2010): 1–10.

“Vision Seaview Gracefield 2030.” *Hutt City Council* (August 10, 2010): 1–38.

“Waiwhetu Stream Hydraulic Modelling 2005-2011” (November 24, 2011): 1–143.

Waldheim, Charles. “Landscape as Urbanism.” In *The Landscape Urbanism Reader*, edited by Charles Waldheim, 35–53. Princeton Architectural Press, 2006.

Waldheim, Charles. “Landscape Urbanism: a Genealogy.” *Praxis* 4 (2002): 10–17.

Ware, Sue-Anne. “Representation, Generation and Emancipation in Virtual Landscapes.” *Landscape Review* 8 (2003): 3–11.

Watts, Laura. *Flood Hydrology of the Waiwhetu Stream*. Resource Investigations, July 27, 2004.

Weinstock, Michael. “Emergence and the Forms of Cities.” *Architectural Design* (March 31, 2010): 1–4.

Weller, Richard, and Martin Musiatowicz. “Landscape Urbanism: Polemics Toward an Art of Instrumentality?.” In *The Mesh Book: Landscape/Infrastructure*, edited by Julian Raxworthy and Jessica Blood, 66–74. Melbourne: RMIT University Press, 2004.

Wigley, Mark. “Conceptual Caretakers.” In *Territorial Investigations*, edited by Annette Balkema and Henk Slager. Rodopi, 1999.

Williams, James. “Deleuze’s Ontology and Creativity: Becoming in Architecture.” *Pli* 9 (October 1, 2001): 200–219.

Wing, Jeannette. “Computational Thinking.” *Communications of the ACM* 49, no. 3 (February 24, 2006): 33–35.

Wiscombe, Tom. *Emergent Processes*, 2009. <http://emergentarchitecture.com/pdfs/OZJournal.pdf>.

Wood, Peter. “Drawing the Line: a Working Epistemology for the Study of Architectural Drawing.” The University of Auckland, 2002.

Woodbury, Robert. *Elements of Parametric Design*. 1st ed. Routledge, 2010.

Woodbury, Robert, Robert Aish, and Axel Killian. “Some Patterns for Parametric Modeling.” In *27th Annual Conference of the Association for Computer Aided Design*, 222–229. Halifax, Nova Scotia: Routledg, 2007.

Wright Steenson, Molly. *Artificial Intelligence, Architectural Intelligence: the Computer in Architecture, 1960–80*, May 10, 2010.

Wright Steenson, Molly. “Problems Before Patterns.” *Interactions* 16 (March 25, 2009): 20–23.

Yannas, Simos. “Adaptive Strategies for an Ecological Architecture.” *Architectural Design* (October 11, 2011): 1–8.

Illustration Credits

1    Reproduced image and quote from “Ivan Sutherland: Sketchpad Demo,”  
accessed October 14, 2012, [http://www.youtube.com/watch?feature=player\\_](http://www.youtube.com/watch?feature=player_embedded&v=BKM3CmRqK2o)  
[embedded&v=BKM3CmRqK2o](http://www.youtube.com/watch?feature=player_embedded&v=BKM3CmRqK2o).

2    Quote from John Frazer in Burry, *Scripting Cultures: Architectural Design*  
*and Programming*, (Wiley, 2011): 51-53. Image reproduced from Frazer, *An*  
*Evolutionary Architecture*, (Architectural Association Publications, 1995): 71.

3    Reproduced from *Arquitectura Viva* 28, (1993): 93.

4    Reproduced from “Yokohama International Port Terminal”, accessed November  
28, 2012, [http://www.arcspace.com/image-library/yokohama-international-](http://www.arcspace.com/image-library/yokohama-international-port-terminal/)  
[port-terminal/](http://www.arcspace.com/image-library/yokohama-international-port-terminal/).

5    Reproduced from McHarg, *Design with Nature*, (San Val, 1995): 111.

6    Reproduced from Spirn, “Ian McHarg, Landscape Architecture, and  
Environmentalism.” In *Environmentalism in Landscape Architecture*, (Dumbarton  
Oaks Research Library and Collection, 2001): 107.

7    Reproduced from “Broken Circle”, accessed March 4, 2013, [http://www.](http://www.robertsmithson.com/earthworks/broken-circle_b.htm)  
[robertsmithson.com/earthworks/broken-circle\\_b.htm](http://www.robertsmithson.com/earthworks/broken-circle_b.htm).

8    Reproduced from “Vietnam Veterans Memorial”, accessed March 4, 2013,  
<http://www.american-architecture.info/USA/USA-Washington/DC-006.htm>.

9    Reproduced from Corner and MacLean, *Taking Measure Across the American*  
*Landscape*, (Yale University Press, 2000): 90 and 83.

10   Reproduced from Tschumi, *Cinnegrime Folie*, (Princeton Architectural Press,  
1987): 3.

11   Reproduced from “Masterplan for the Parc de la Villette in Paris”, accessed  
March 16, 2012, <http://oma.eu/projects/1982/parc-de-la-villette>.

12   Reproduced from “Flowing Gardens, Xian, China, International Horticultural  
Fair, 1st Prize”, accessed December 6, 2012, [http://groundlab.org/portfolio/](http://groundlab.org/portfolio/groundlab-project-flowing-gardens-xian-china/)  
[groundlab-project-flowing-gardens-xian-china/](http://groundlab.org/portfolio/groundlab-project-flowing-gardens-xian-china/).

13   Reproduced from “JAMES CORNER FIELD OPERATIONS AND  
DILLER SCOFIDIO + RENFRO”, accessed December 11, 2012, [http://www.](http://www.thehighline.org/james-corner-field-operations-and-diller-scofidio-renfro)  
[thehighline.org/james-corner-field-operations-and-diller-scofidio-renfro](http://www.thehighline.org/james-corner-field-operations-and-diller-scofidio-renfro).

14   Reproduced from “Fresh Kills Park Project”, accessed December 11, 2012,  
<http://www.nyc.gov/html/dcp/pdf/fkl/fied2.pdf>.

15   Reproduced from Woodbury, *Elements of Parametric Design* (Routledge, 2010):  
203.

16   Reproduced from Woodbury, *Elements of Parametric Design* (Routledge, 2010):  
204.

17   Reproduced from Alexander et al, *A Pattern Language*, (Oxford University Press,  
1977): 544.

18   Reproduced from Alexander et al, *A Pattern Language*, (Oxford University Press,  
1977): 515.

19   Reproduced from Alexander et al, *A Pattern Language*, (Oxford University Press,  
1977): 518.

20   Reproduced from Alexander et al, *A Pattern Language*, (Oxford University Press,  
1977): 463.

21   Image by the author.

22   Image by the author.

23   Image by the author.

24   Image by the author.

25   Image by the author.

26   Image by the author.

27   Image by the author.

28   Image by the author.

29   Image by the author.

30   Image by the author.

31   Image by the author.

32   Image by the author.

33   Image by the author.

34   Image by the author.

35   Image by the author, adapted from Google Maps imagery.

36   Image by the author, adapted from Google Maps imagery.

37   Image by the author.

38   Image by the author.

39   Image by the author.

40   Image by the author.

41   Image by the author.

42   Image by the author.

43   Image by the author.

44   Image by the author.

45   Image by the author.

46   Image by the author.

47   Image by the author.

48   Image by the author.

49   Image by the author.

50   Image by the author.

51   Image by the author.

52   Image by the author.

53   Image by the author.

54   Image by the author.

55   Image by the author.

56   Image by the author.

57   Image by the author.

58   Image by the author.

59   Image by the author.

60   Image by the author.

61   Image by the author. Image by the author.

62   Image by the author.

63   Image by the author.

64   Image by the author.

65   Image by the author.

66   Image by the author.



