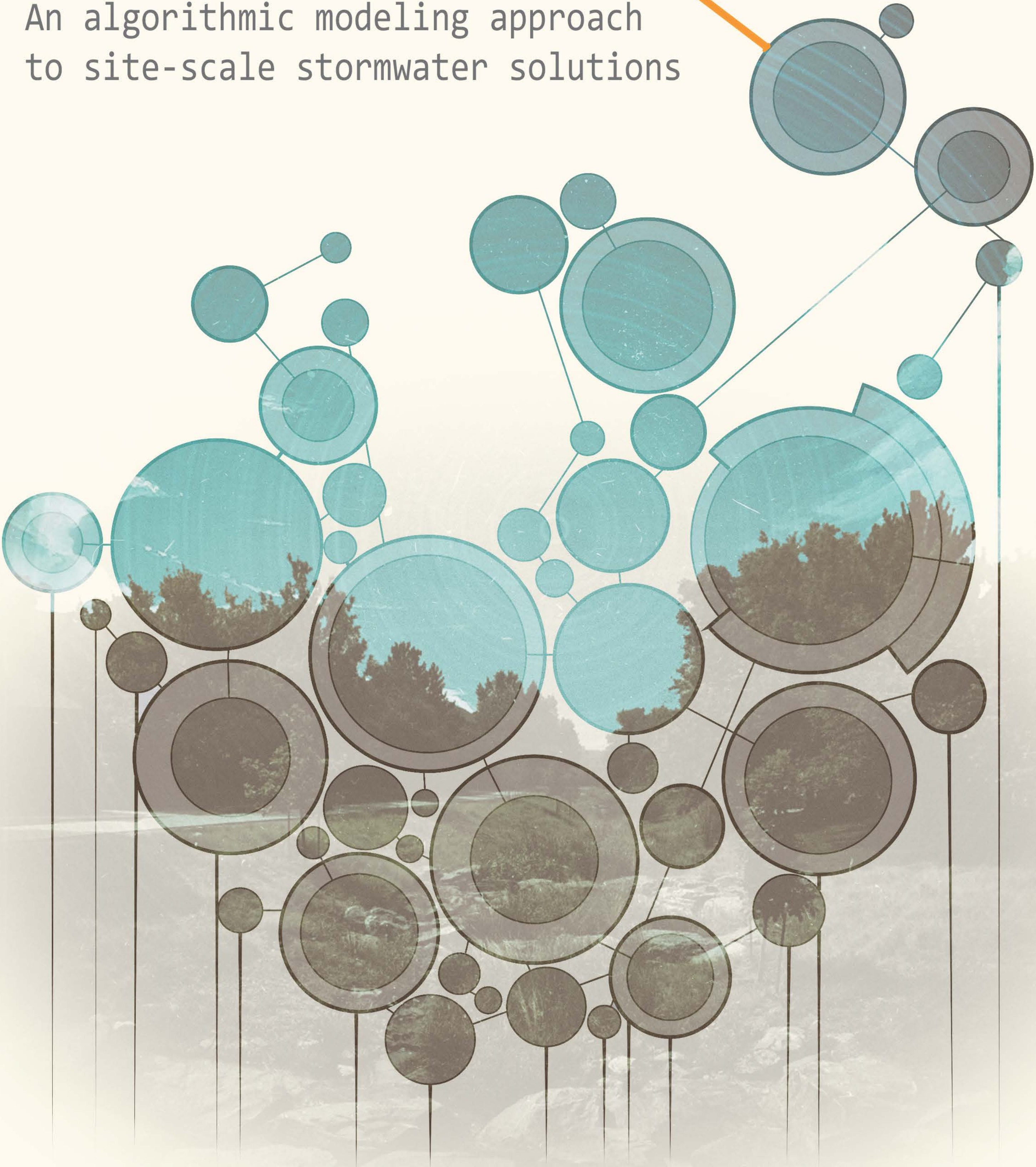
DYNAMIC DITCHES

An algorithmic modeling approach to site-scale stormwater solutions



Dynamic Ditches:

An Algorithmic Modeling Approach to Site-Scale Stormwater Solutions

Christopher G Weaver, 2019

Approval

DYNAMIC DITCHES :An Algorithmic Modeling Approach to Site-Scale Stormwater Solutions

by

Christopher G Weaver

Project Chair **Dr. Mark Eischeid**

Committee **Dr. Chris Enright**

Submitted in partial fulfillment of the
Master of Landscape Architecture
Department of Landscape Architecture
University of Oregon
June 13th, 2019

Acknowledgements

This document would not have been possible without the guidance and support of so many. I owe thanks to the following:

First and foremost, I owe nearly everything to Virlynda. Your unwavering support throughout this project—and the entire program—was nothing short of stunning. Without it, I would not have become so totally awesome as I undoubtedly am now.

Isla—you are the reason why I do anything. Watching you grow and learn about this strange world has pushed me to care about others more than I ever have.

Dr. Mark Eischeid—I cannot thank you enough for your mentorship, all of your crucial feedback and the many opportunities that ensured my success in this program.

Dr. Chris Enright—your commitment to your students inspired me, and you provided the guidance and structure that I desperately needed.

Jun Hak Lee—thank you for your expert guidance environmental modeling. I aspire to your level of knowledge and proficiency.

Everyone at GreenWorks—your enthusiastic support of a student project is a credit to your dedication to the profession. Thank you for your encouragement, advice and the resources.

Abstract

This project sought to explore ways in which algorithmic modeling may lend a methodology to the landscape architecture design process. Specifically, a computational design framework originating from industrial engineering was applied to the problem of siting and sizing bioretention facilities—a type of stormwater infiltration facility—on a site-scale. The primary task was in developing the algorithm in Rhino+Grasshopper to automate the logic sequence and calculations of the standard multi-step stormwater design procedure. There were several essential goals in this research project: 1) broaden the range of stormwater solution options for landscape architects, 2) develop the capability to optimize design solutions for low-cost and localized impact, 3) create an algorithmic workflow that may be expanded upon to encode additional landscape design variables.

As a measure of validity, the completed algorithm was tested against a professional design for the 0.8 acre LaTourette Park in Oregon City, OR. Both were evaluated based on their performance in a 10-year design storm. Compared with the professional stormwater scheme, the algorithmic model resulted in more efficiency in terms of lowering construction costs, percentage of runoff reduction, and in intercepting surface flow near its source. These are ostensibly favorable results, although the utility of the algorithmic model is debatable. Being a simulation narrowly geared toward a single type of solution, the model does not offer solution alternatives that may involve a broader stormwater management scheme. However, it may serve as a tool in informing decisions of that kind.

Table of Contents

List of Figures

List of Tables

“In order to design for movement, a whole new system of conceptualizing must be undertaken. Our present systems of design and planning are invariably limited by our techniques and our methods of symbolizing ideas, so that is all we do.”

—Lawrence Halprin[1]

1 Halprin, Lawrence. *Cities*. (Cambridge: MIT Press, 1972): 5

1 Introduction

1.1//Computational Design for Landscape Architects

In the current state of design, the methods of abstracting the world are becoming increasingly computational and data driven.[2] This fact underlies the imperative for landscape architects to innovate their design approach by increasing their influence in the digital realm and embracing approaches rooted in computational design. Doing so offers more than simply upgrading tools for executing typical computational tasks. Computational design offers a means of changing the way landscape architects understand their design process. Perhaps counter-intuitively, the very process of computational design is less about directly determining a design form and more about defining and controlling a *design environment*[3]—in which a system of parameters and rules automatically produce design solutions. This approach provides a wholly different means of understanding landscape architecture, and it does so in such a way that goes beyond static modes of analysis and visual representation. The designer undertakes a process of abstraction that requires an explicit—albeit simplified—definition of the complex interactions occuring in the landscape. From that abstraction the designer devises a set of rules for automatically outputting changes that occur in the design. These programmed responses are therefore characterized as *generative*—as opposed to being purely subjective. With this type of approach, the act of design

2 Cantrell, Bradley & Mekies, Adam. "Coding Landscape." In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. (New York, NY: Routledge, 2018), 5 -33.

3 *Design environment* in this usage refers to the digital context in which the design is being represented.

comes closer to reflecting the same level of dynamism as the very systems that are being designed and studied in landscapes.

Adopting computational design techniques can provide a range of improvements in routine design tasks and opens up new capabilities. No doubt this process is technically complex, and requires from the designer a facility with workflow logic and coding language that is part-and-parcel of computational design. The algorithmic mechanism of this design strategy forces an explicit definition of concepts that the designer may grasp intuitively. Although, arguably, the educational utility of computational design is to advance the critical thinking of the designer by enhancing the means of generating design solutions.[4]

In this project, I seek to explore the utility of the computational design approach to the design of landscapes, using it as a means of handling a common landscape design problem—stormwater management. Stormwater management is a fundamental aspect of any landscape design. The design of a stormwater management scheme simultaneously factors aspects of ecology, hydrology and human comfort—the improvement to each being core objectives in creating any quality, sustainable built environment. The design of stormwater management systems exemplifies the more complex challenge of landscape design by offering opportunities to synergize the many systems in play.

The end-product of this project is a Rhino+Grasshopper[5] algorithmic model which automates the more tedious aspects of stormwater design, while also offering the designer expanded capabilities in exploring some of the more complex aspects of design. The algorithmic model is potentially a generic design tool—in that it may be easily adapted to produce designs for any other site, as long as the appropriate input data is provided. The utility of this approach is demonstrated directly in the design of stormwater solutions for the Latourette neighborhood park in Oregon City, Oregon.

4 Caliskan, Olgu. "Parametric Design in Urbanism: A Critical Reflection." *Planning Practice and Research*. Vol. 32 No. 4 (2017): 417-443.

5 Rhino is a 3D modeler released by McNeel used to create, edit, analyze, document and render digital 3D geometry. The plugin Grasshopper is the interface for McNeel's proprietary visual coding language used to construct algorithms executed by the modeler.

1.2//Systems Theory in the Built Environment

To best understand the intersection of landscape architecture and computational design, it is necessary to appreciate the theoretical basis of systems thinking. The following section outlines the essential history of this field of study where it pertains to the contemporary understanding of computational design for the built environment.

i. From Weapons to Landscapes

Surprisingly, systems thinking in design has its origin in the military-industrial complex of World War II.[6] One figure in particular, Warren Weaver, is an important linkage between the wider realm of academia and the wellspring of technological advancement that was the 20th century arms race. Under the executive directive of President Franklin Roosevelt, Weaver started the Office of Science Research and Development (OSRD) in 1941. The explicit purpose of this newly created federal agency was to advance the military's technological capabilities and, having been created in the midst of wartime, had nearly limitless resources at its disposal.[7]

The most important product of this agency came directly from the work of Weaver himself—that of fire control systems for guiding artillery bombardment. Specifically, the agency managed to integrate the SCR-584 radar array with the M9 artillery control and developed the capability for its remote operation. This innovation marked the first time that the US military regarded its equipment as a weapons system instead of as individually operated objects.[8] A notable offshoot of this research was the literature that would become the foundational text in the study of computing and information theory. *Cybernetics: Or Control and Communication in the Animal and the Machine* (1948) by OSRD mathematician Norbert Weiner posited a mathematical basis between the "signals, feedback, and control" that was seen to exist in both man-made and natural systems. Weaver continued in this vein of research of information systems, proposing a framework for classifying systems analyses according to degrees of complexity.[9]

He postulated that scientific problems can be generally categorized in one of three types, each with its own appropriate method of informational organization. *Problems of simplicity* largely encompassed the scientific and technological advances that led up to the contemporary era. These were two-variable problems that were easily observed, quanitfied and manipulated. On the other end of the spectrum of problems of simplicity were *problems of disorganized complexity*. These regarded systems with many concurrent variables and an unobservable quantity of elements. Despite their cumbersome size, these systems

6 De Monchaux, Nicholas. *Local Code: 3,659 Proposals about Data, Design & the Nature of Cities*. (New York, NY: Princeton Architectural Press, 2016) 127.

7 De Monchaux, *Local Code*, 129.

8 Ibid, 130.

9 Weaver, Warren. "Science and Complexity." *American Scientist*. no. 36 (1948): 536.

could be studied and understood through statistical analyses. The middle designation in Weaver's spectrum were *problems of organized complexity*, considered too large for simple mathematical analysis but too small for statistical methods. In the many years since the publishing of these works, entire branches of systems theory have delved into the prediction of systems exhibiting organized complexity—a type of system that has consistently been problematic and difficult to nail down. However, it was not until the publication of *Cybernetics* that problems such as urban planning were understood as problems of organized complexity.

Cybernetics, the understanding of communication and control in systems, was employed to reduce the intricately interrelated variables of urban systems into a multivariate system of equations.[10] For instance, in 1962 the RAND Corporation conducted an urban planning study and produced their "Model of a Metropolis" urban planning model (Figure 1.1) from which the city of Pittsburgh could purportedly base its comprehensive planning decisions.[11] In retrospect, these models were overly simplistic and coincided with the authoritatively-bent urban renewal movement that defined Modernist urban planning. The movement is characterized as having had a unilateral approach to the design of cities that greatly underestimated the incomprehensibility of the interconnected urban environment—instead offering blanket solutions to what was perceived as urban blight.

It is partly a backlash to this movement that helped to diversify systems theory's application in the realm of design. Most renowned is Jane Jacob's activism against Robert Moses' sweeping urban renewal schemes across New York City. Her polemic against the movement, *Death and Life of Great American Cities* (1963), was largely based on her own observations of the great multitude of momentary human interactions in the urban setting. She astutely observed that people's collective behavior followed a pattern that, while broadly predictable, was hugely complex with an unquantifiable degree of interpersonal and systemic interactions. Jacobs' work refutes the notion that urban organized complexity is either definable or controllable, arguing that the top-down urban planning approach completely fails to capture the essence of human responses to urban systems.[12] Still, cybernetics dominated in systems theory for decades.

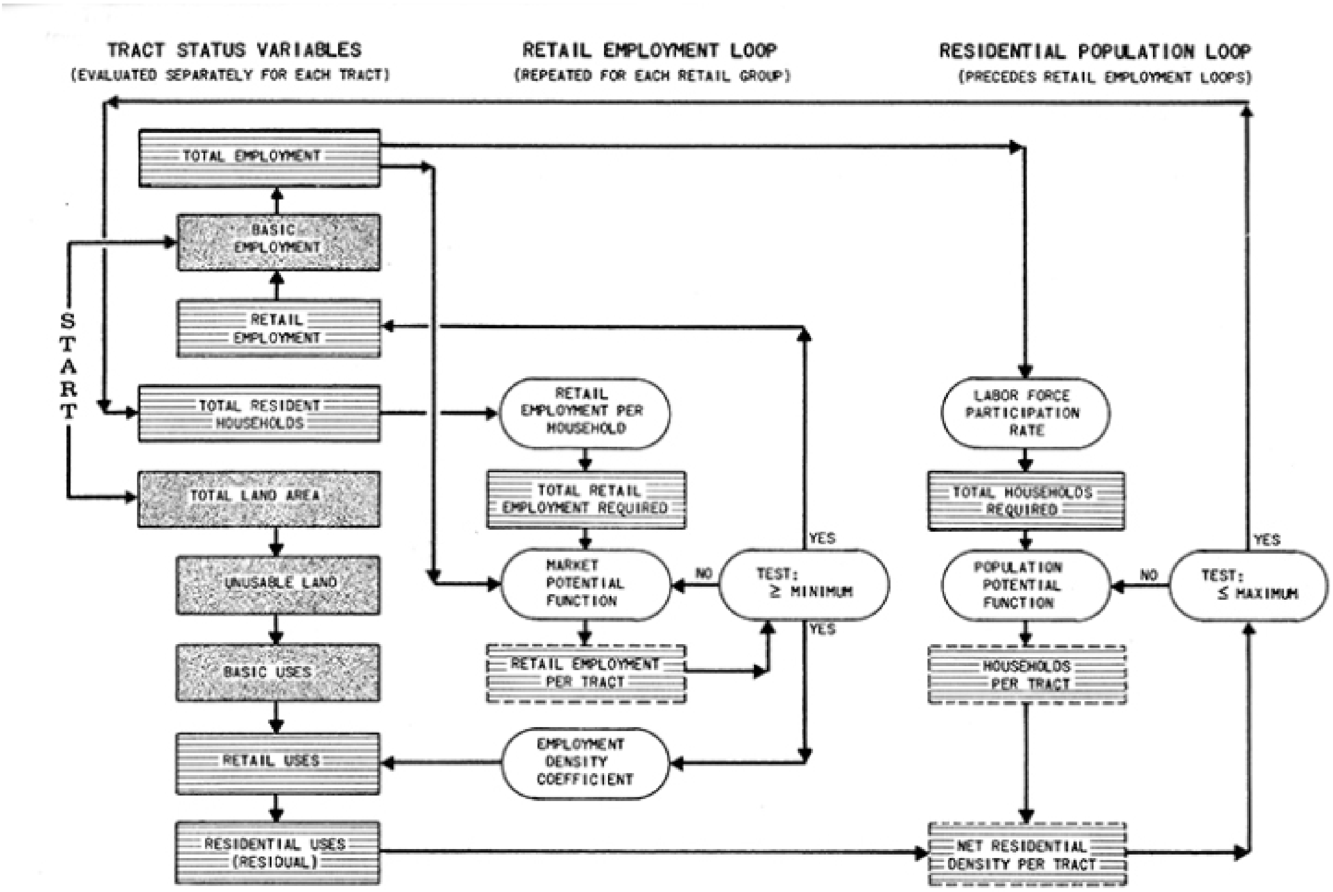
Overlay mapping was another development which can trace a linkage to the research and figures of cybernetics. What would eventually become Geographic Information Systems (GIS)—a standard tool of landscape architecture—started as the Semi-Automated Ground Environment (SAGE) air-defense targeting system based upon the fire control systems developed by the OSRD.[13] By the mid-60's, the principles for the operation of this weapons system had developed into a visualization method in landscape planning and analysis. First

10 Forrester, Jay. *Urban Dynamics*. (Cambridge, MA: MIT Press, 1969): 12.

11 De Monchaux, *Local Code*, 133.

12 Jacobs, Jane. *Death and Life of Great American Cities*. (New York: Vintage Books, 1963): 405.

13 De Monchaux, *Local Code*, 159.



//Figure 1.1

Overview of informational flows to guide urban planning and policy decisions. (RAND Corporation. Accessed May, 20th 2019. https://www.rand.org/content/dam/rand/pubs/research_memoranda/)

through the efforts of Howard Fisher and Carl Steinetz—both affiliated with Harvard Graduate School of Design—the SYMAP computer system was developed for the explicit purpose of integrating mapping systems and for cartographic plotting.[14] Further collaboration between Steinetz and Ian McHarg, of the University of Pennsylvania Department of Landscape Architecture, applied these graphic techniques as a decision-making design tool for landscape planning.[15] Similar to the cybernetic approach in urban planning, the overlay analysis (aka. McHargian Analysis) is a top-down technique that reduces options though a series of binary choices. Overlay analysis continues as a common method in landscape architecture today, albeit with an awareness of its propensity to weigh towards the designer's biases.[16]

ii. Generative Systems

Contemporaneous to the developments in overlay analysis, a different type of inquiry in systems thinking was being made by architect and design theorist Christopher Alexander. A decade before his best known book *A Pattern Language* (1978), Alexander argued that there is a key distinction to be made in conceptualizing systems.[17] In one sense, he argues, *systems as a whole* are a way of viewing an object as a product of the interaction of all its component parts. It is viewing a thing holistically and giving definition to the relationships between components which give meaning to the entire assemblage. In another sense, Alexander introduces the idea of a *generating system*, or the rules for the relationships between components that give rise to any system. Alexander notes that these concepts are superficially similar, but the difference is that the "system as a whole" aspect examines a single outcome of a generating system which provides a logical structure capable of producing any number of different outcomes.[18] This branch of systems theory is markedly different from the cybernetics theory which emerged from the results-oriented military research apparatus. In terms of design and planning, it does not easily arrive at prescriptive outcomes and instead focuses on understanding the logic of processes within a system.

Continuing in this vein, architect and theorist Stan Allen proposed in "From Object to Field: Field Conditions in Architecture and Urbanism"(1997) his concept of *field conditions* as a way of characterizing quantifiable elements in the design environment. Field conditions are defined as "...any formal or spatial matrix capable of unifying diverse elements while respecting the identity of each."[19] Similar to the reasoning of Alexander's generating systems, fields are a means of perceiving a

14 Cantrell & Mekies, "Coding Landscape", 7.

15 Ibid, 8.

16 Girot, Christophe. "About Code." In *Codify: Parametric and Computational Design in Landscape Architecture.* Edited by Bradley Cantrell and Adam Mekies. (New York, NY: Routledge, 2018), 1 – 4.

17 Alexander, Christopher. "Systems Generating Systems." *Architectural Design*. no. 38 (1968): 605.

18 Ibid, 607.

19 Allen, Stan. "From Object to Field Architecture and Urbanism." *Architectural Design*, no. 127 (1997): 24-31.

gestalt characterization from the collective qualities of a given system—that being a result of the perceptible processes among component parts. It is an inherently bottom-up perspective which starts with an examination of the localized quality of an element and then scales upward to examine the cumulative quality of a collection of those elements.

A classic example is the emergent formation of flocks from the individual flight patterns of birds.[20] A single bird may only be aware of its immediate surroundings, controlling its flight merely to avoid colliding with its neighbors while staying within a certain distance. The result of hundreds or thousands of birds following those same rules is a flock that appears to be controlled by a shared consciousness. In actuality, the flock's behavior is simultaneously random but collectively controlled, in that variations in any individual's behavior can create a cascade of reactions throughout the entire flock.

Field conditions, as a theory, rejects the notion that systems inherently have hierarchical ordering and imposed rules. Instead, it promotes fluidity and a responsiveness to localized condition.[21] Field conditions have had a considerable impact on the development of computational design in landscape architecture. Describing phenomena in terms of field conditions sets up the computational definitions that better reflect the dynamism of landscape systems, especially when looking beyond strict spatial orderings. Thinking and designing in terms of field conditions reframes conceptions of landscape systems and their visualizations. Mapping field conditions, as opposed to static conditions, produces patterns that vary with intensity—where focal points emerge as peaks and valleys from the depicted field.[22] And in the sense of coding algorithms and simulations, the perception of field conditions is vital to the abstraction of phenomena within a design into computational representations.

20 Ibid, 26.

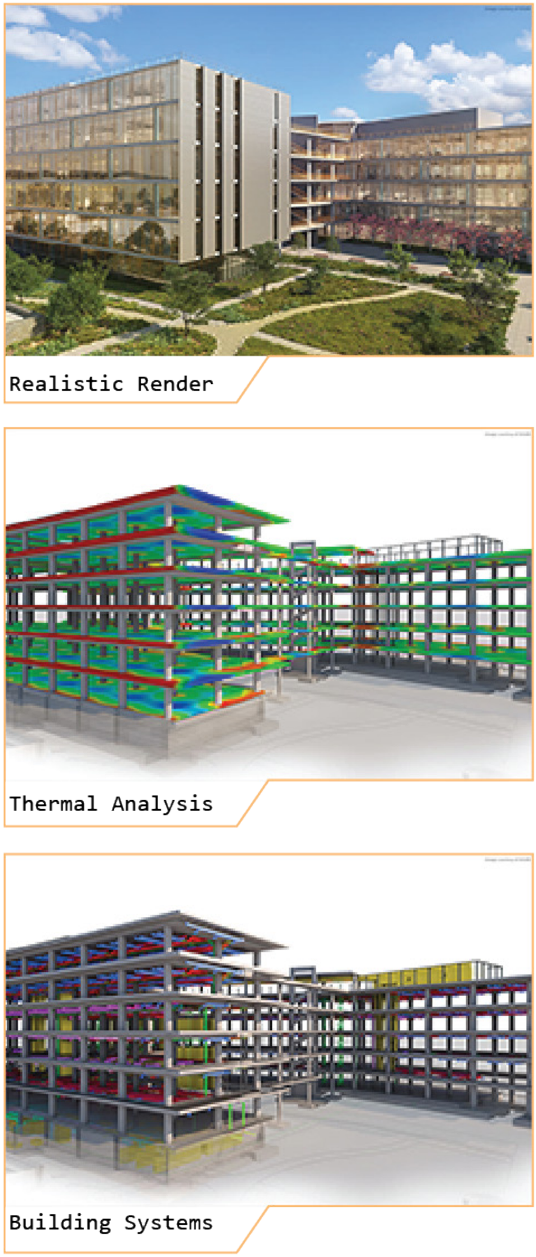
21 Belesky, "Field Conditions," (Groundhog plugin website. Accessed October 22nd, 2018, https://www.groundhog.la/techniques/field-conditions/).

22 Allen, "From Object to Field." 27.

1.3//State-of-the-Art of Computational Design

Historically, the main utility of digital tools in design has been to support preconceived designs with enhanced graphics and organizational abilities. However, with new capabilities and advances in processing power, computers are increasingly an artificial extension of our natural capacity to understand context and engage in critical thinking during the process of design.[23] Ever-improving sophistication in technology is leading to increasingly useful digital tools which can more accurately simulate real-world phenomena.

The focus of this research project—algorithmic modelling—significantly assists the design process by encoding systemic responses to a quantity and diversity of inputs that are too numerous for the human mind to reasonably handle. In its most common usage, algorithmic modelling is specialized at form-making and simulating the performance of built structures.[24] Figure 1.2 gives a glimpse of that functionality, as it exists in the architectural software Revit. In its theoretical full potential, algorithmic modeling can be a sophisticated generative tool from which highly complex designs of the entire built environment emerge from explicit defini-



//Figure 1.2

Current application of computational design in the field of architecture. Revit screenshots of different layers of the same design show how design functionality is paired with visual representation. (Revit website. Accessed June 9th, 2019. https://www.autodesk.com/products/revit/overview)

23 Christoforetti, Elisabeth; Cohen, Will; Cohen, Yonatan; Rife, Stephen; Zhang, Jia. “Big Data for Small Places.” In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. New York, NY: Routledge (2018): 5-33.

24 Belesky, Philip. “The Green Grasshopper.” *Journal of Digital Landscape Architecture*, no. 3 (May 30, 2018): 406–413.

tions of design rules.[25]

The use of algorithmic tools such as Revit+Dynamo, Catia and Rhino+Grasshopper has been limited to the purview of experts only until the past decade or so.[26] And within the design professions, the technology has been most heavily utilized and researched in architecture and engineering. In landscape architecture, computational design has a proportionally smaller application in academia and private practice. Nonetheless, innovators are discovering the potential of the technology to realize the discipline's agenda of responsiveness, flexibility, and adaptability.

The following section examines a chronological series of projects from across different design disciplines that reflect either major developments or the current state-of-the-art of computational design that impact the field of landscape architecture.

25 Caliskan, "Parametric Design in Urbanism: A Critical Reflection," 430.

26 Girot, "About Code," 2.

Project Name | Location: Kartal-Pendik Masterplan | Istanbul, Turkey
Project Type: Master planning competition
Design Firm | Designer: Zaha Hadid Architects | Patrik Schumacher
Year Completed: 2006

Zaha Hadid Architects (ZHA) is arguably at the forefront of computational design, having been developing it as a specialty for the last fifteen years. Before winning the design competition for the Kartal-Pendik masterplan, the firm had employed its signature style, Parametricism, to win three high profile international master planning competitions. Developed by partner Patrik Schumacher, Parametricism is posited as both an architectural style and a design theory. This project was a design proposal for a 136-acre city subcentre on Istanbul's Asian side of the Bosphorus with the goal of reducing pressure on the European side.

The design would redevelop a post-industrial area of Istanbul and become a link between Asia and Europe. The area was treated as a blank slate, allowing ZHA to utilize parametric design techniques to determine the spatial composition via road networks throughout the design area.[27] Final designs were derived from the algorithmic process of making thread-like connections between existing street networks on the site periphery (Figure 1.3.a). The finished render of the plan (Figure 1.3.b) displays the outcome of that technique—the distinct organically shaped street network and the seemingly warped figure-ground aspect of the resultant building layout.

Despite the international acclaim that this design has garnered, the project remains in the conceptual stage, with no indication of actually being built.[28] The main reason for this is that the model did not incorporate existing city codes. And when the plan advanced to the post-concept phase, it proved to be inflexible to such a fundamental change in the algorithmic definitions.[29] Therefore a vital takeaway from this project is that the algorithmic modeling process must be open to the subject judgements that are separate from automated outcomes. If the algorithm itself is unable to incorporate important factors such as feasibility and statutory regulation, then the designer controlling that algorithm must interpret those conditions by other means.

27 Carpo, Mario. The Digital Turn in Architecture 1992-2012. (West Sussex: John Wiley & Sons Ltd. 2013): 22.

28 Ibid, 241.

29 Caliskan. "Parametric Design in Urbanism: A Critical Reflection," 432.

a//Street connections

b//Final plan rendering

//Figure 1.3

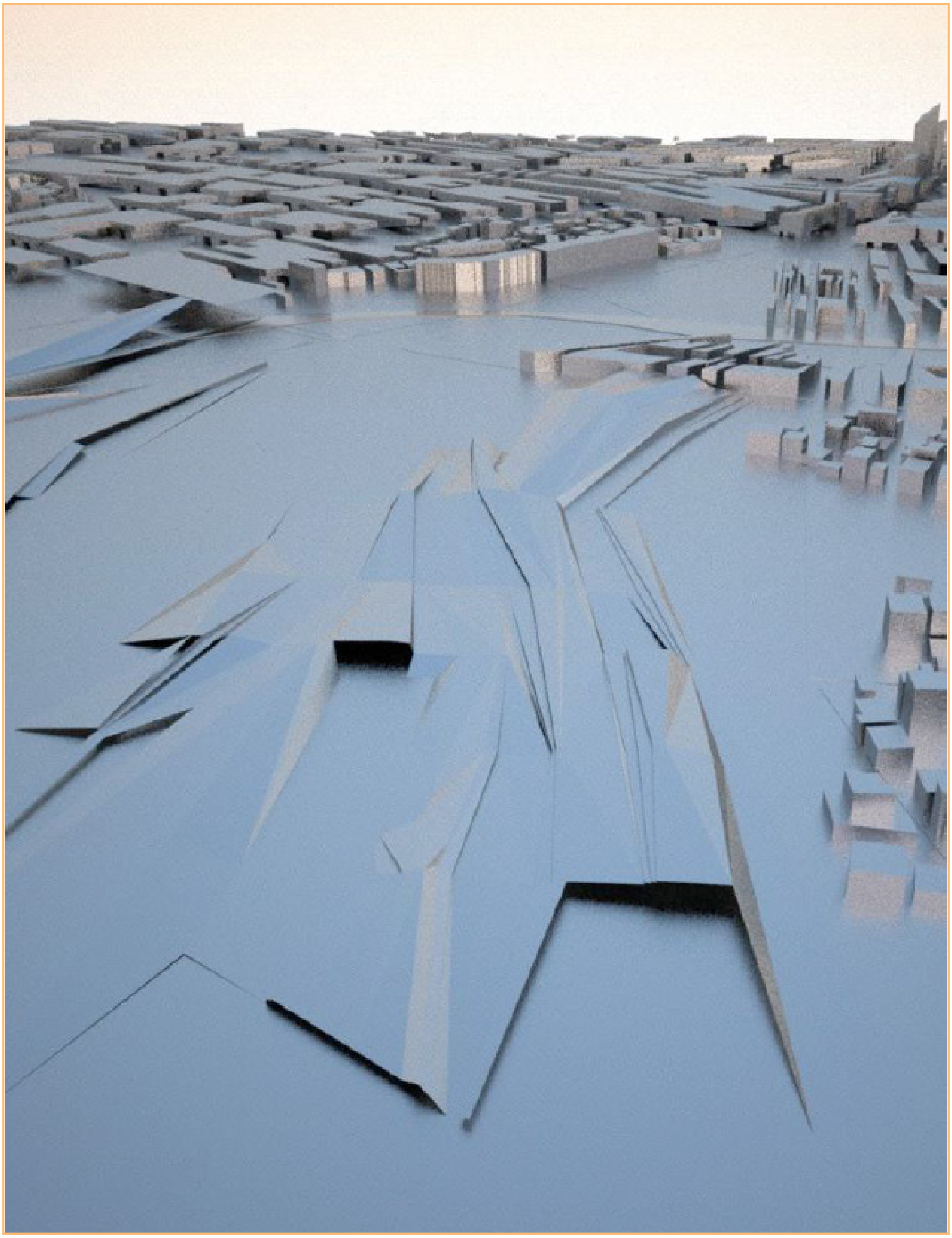
(a) Process drawing of the algorithmically determined street network connections in Kartal-Pendik masterplan and (b) final plan rendering. (Zaha Hadid Architects, 2006; https://www.zaha-hadid.com/masterplans/kartal-pendik-masterplan)

Project Name | Location: Deep Ground | Shenzhen, China
Project Type: Master planning competition
Design Firm: Groundlab
Year Completed: 2008

This international master planning competition winner aimed to revitalize 11.8 km^2 urban land, primarily by creating an ecological corridor. Simultaneously the design aimed to pattern urban infill after the character of existing neighborhoods immediately outside of the development area. In a move that proved innovative and successful for the time, Groundlab programmed an algorithmic model that created exaggerated variations in the natural terrain along the proposed corridor which, in turn, determined building configurations. Their method—which they called *thickened ground* (Figure 1.4)—created a three-dimensional spatial complexity which had not been adequately developed in past computational models.[30] The model uses surface operations such as folding and leveling to create distinct pedestrian levels with irregular variations in depth and size.

The model also successfully managed to create a catalogue of generative building typologies which met project goals for matching existing patterns of the villages. Along with a subtle internal variation of traditional courtyards and open spaces, the algorithmic model generated alterations of the urban layout from parameters such as the proximity of density nodes.

30 Ibid, 429.



//Figure 1.4

Model rendering displaying thickened ground variations along a proposed ecological corridor in the Shenzhen master plan. (Groundlab, 2008; http://groundlab.org/)

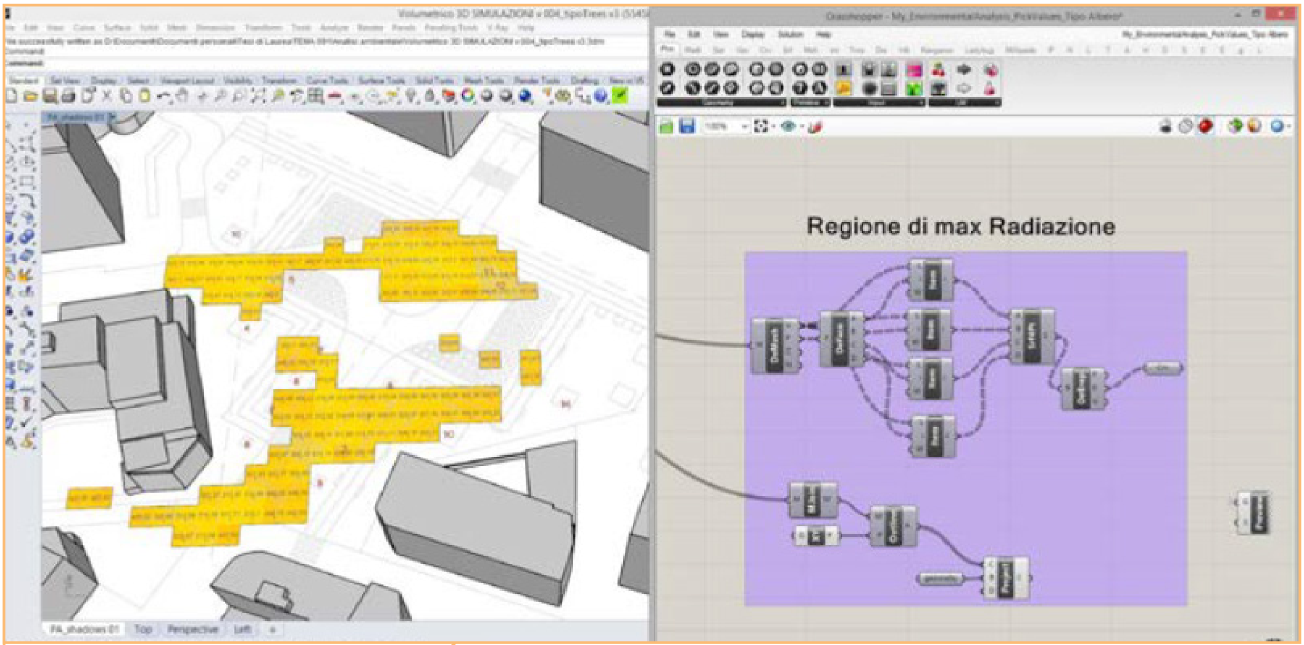
Publication: "Parametric environmental climate adaptive design"
Project Type: Research of urban climatic resiliency
Researchers: Eduardo Bassolino & Luciano Ambrosini
Year Completed: 2015

This study[31] utilized computational modeling tools to simulate the effects of climate change on an existing public plaza in historic Naples, Italy. The objective was to devise long-term resiliency strategies to maintain a level of human comfort throughout changing conditions. Within the simulation, the researchers managed to accurately analyze microclimatic conditions as a function of both weather and the material/spatial properties of the plaza. Both current conditions and projected climatic conditions were simulated in order to determine constraints and performance targets for basic design solutions.

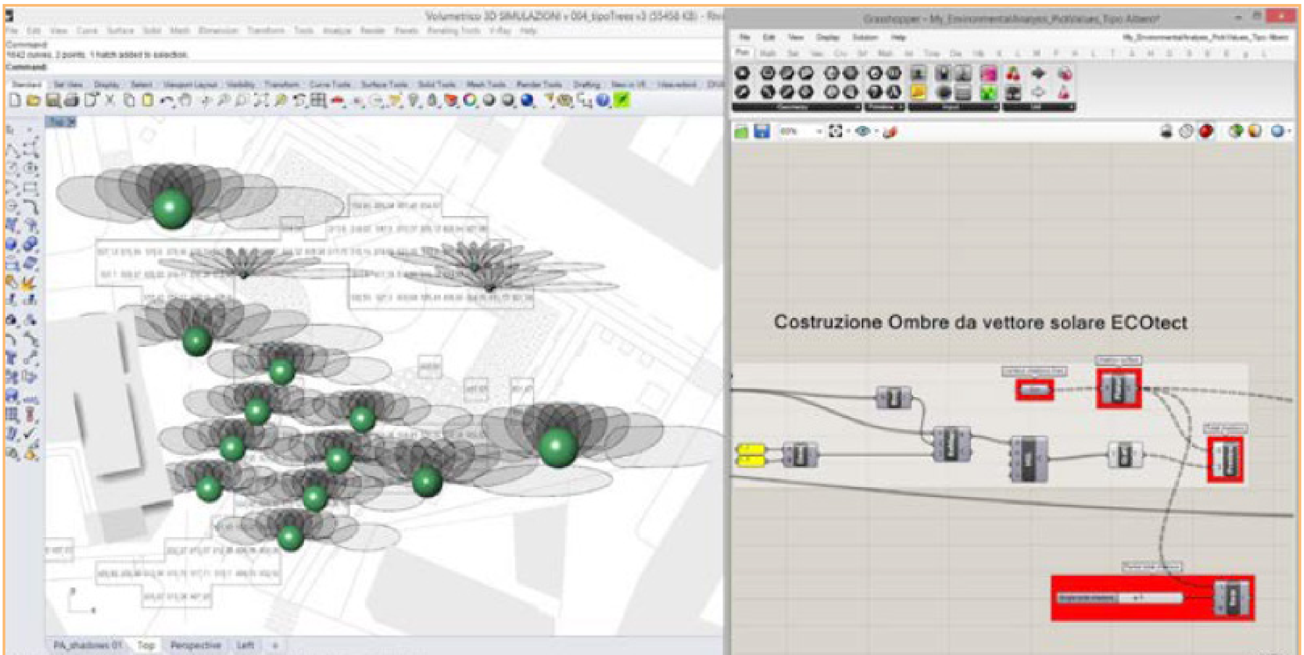
First, a model of the area was built in Rhino that included certain microclimatic properties, such as incident solar radiation, insolation sunlight hours and weather data (Figure 1.5.a). The model was then imported into a different analysis software, Envi-MET 3.1, in order to better simulate the complex interactions of climatic factors and output data in terms relatable to human experience, Physical Equivalent Temperature (PET). This data was imported back to the Rhino model and spatially related along a high-resolution grid. This served as a baseline against which researchers could measure design solutions.

The design solutions were simply morphological changes to the surface of the plaza, testing different configurations of vegetated areas and added shade trees. Several performance metrics determined the outputs of an evolutionary generative model (Figure 1.5.b). Researchers parametrically altered performance targets which the algorithm attempted to achieve by systematically testing extremes in values, gradually narrowing in on optimal values.

31 Bassolino, Eduardo & Ambrosini, Luciano. "Parametric environmental climate adaptive design: the role of data design to control urban regeneration project of Borgo Antignano, Naples." *Social and Behavioral Sciences*. No. 216 (2016): 948-959.



a//Climatic stress analysis



b//Shade tree placement

//Figure 1.5

(a) Rhino+Grasshopper simulation of climatic stresses on the Borgo Antignano piazza and (b) the generated designs of shade tree placements. (Bassolino & Ambrosini, “Parametric environmental climate adaptive design,” 952.)

Project Name | Location: MAX IV Laboratory | Lund, Sweden
Project Type: Defensive topography
Design Firm : Snøhetta
Year Completed: 2016

A major concern during the design of the MAX IV Laboratory was the potential disruptions to extremely sensitive laboratory equipment due to vibrations from the nearby roadways.[32] In response to this unique constraint, designers conceptualized a topographic wavefield pattern to attenuate and disperse surface vibrations. Precision was a key factor for the efficacy of this design. The performance target was to dampen 10–40m wavelengths with a 4.5m amplitude.[33]

In order to arrive at an optimal configuration, the algorithmic model needed to enable continuous testing of experimental conditions provided by a collaborating engineering team. The morphological components—the spiral wavefield of gently sloping mounds—and their effects were quite literally a field condition manifested in a landscape design.

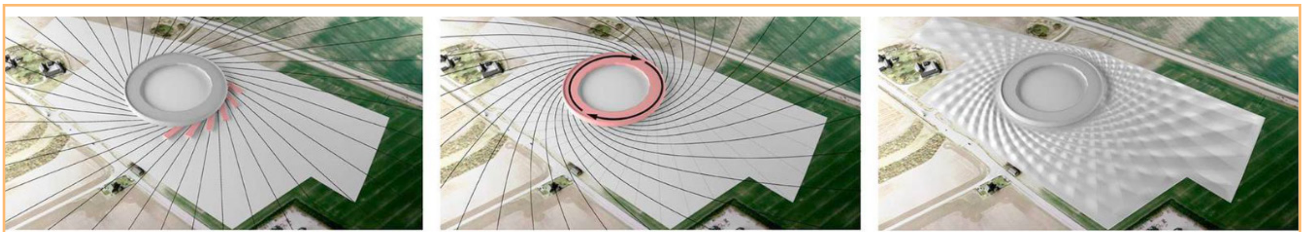
Algorithmic definitions for arc lengths, radii and mound dimensions were adjusted according to performance outputs directly incorporated into the model. Process stage renders of the wavefield design can be viewed in the middle figure of Figure 1.6. The basic geometric pattern is formed from the overlap of opposing spirals extending tangentially from the circular structure of the laboratory. Resulting topographies were analyzed and evaluated based on secondary design criteria such as maximum slope gradients and stormwater management.[34] The project sets a standard in landscape architecture for precision obtained through multivariate simulation.

32 Walliss, Jillian, and Rahmann, Heike. *Landscape Architecture and Digital Technologies : Re-conceptualizing Design and Making*. (Abingdon, Oxon [UK] ; New York: Routledge, 2016): 45 - 68.

33 “MAX IV Laboratory Landscape,” (Snohetta website. Accessed October 2018. https://snohetta.com/projects/70-max-iv-laboratory-landscape)

34 Ibid.

a//Ground perspective



b//Generation pattern

c//Aerial perspective

//Figure 1.6

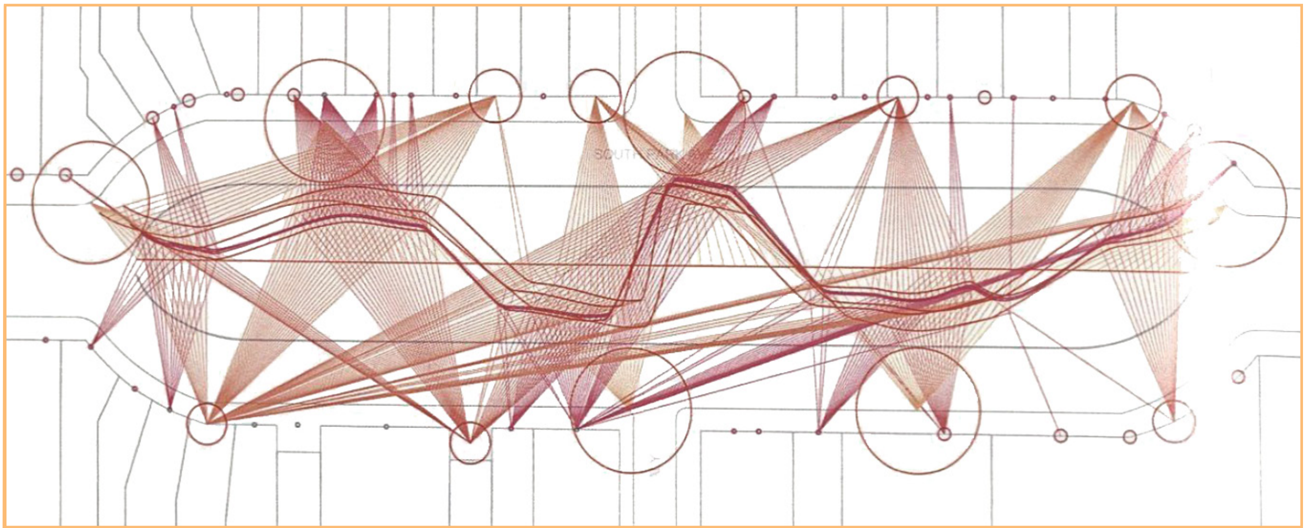
(a) Ground-level perspective of MAX IV Lab topographic design. (b) Simplified process sequence for wavefield pattern and (c) aerial perspective. (Snøhetta, 2016. https://snohetta.com/projects/70-max-iv-laboratory-landscape)

Project Name | Location: South Park | San Francisco, CA
Project Type: City park improvement
Design Firm | Pricipal Designer: Fletcher Studio | David Fletcher
Year Completed: 2018

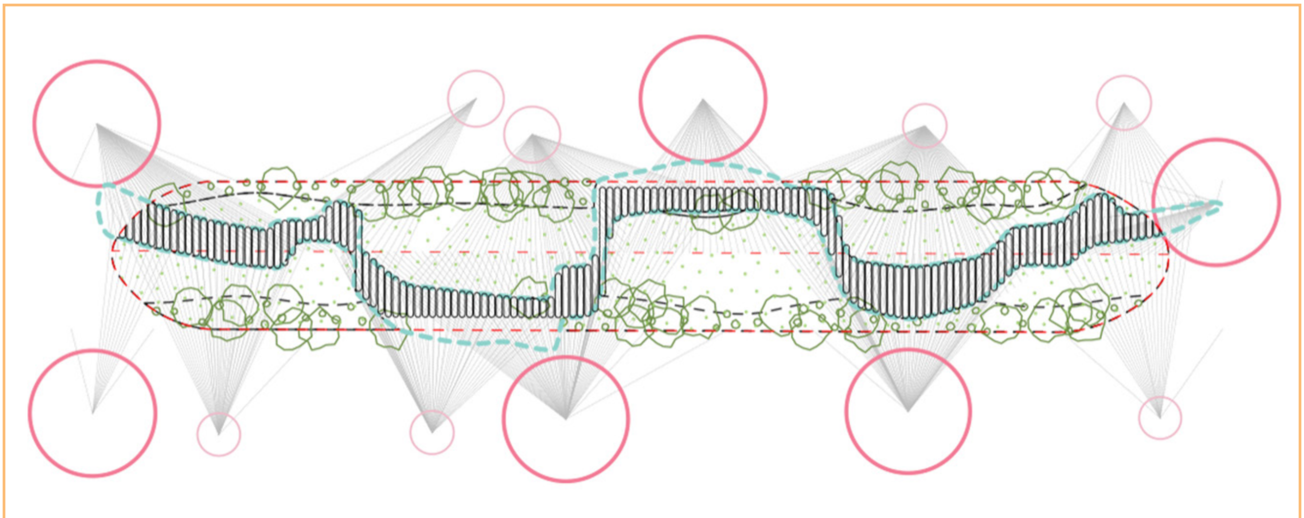
This recent project by Fletcher Studio was created through a hybridized process of iterative analog diagramming and algorithmic modelling in Rhino+Grasshopper. Partly an experiment in computational design, the team approached the project with the stated goal of developing a continuous and responsive morphological operation for determining a modular paving pattern.[35] Early stages of site analysis and conceptualization were performed with typical analog design techniques. These steps informed the algorithmic encoding of relationships between spatial and morphological elements.

Primarily the model was used to generate paving patterns and configurations in response to variations in an interrelated series of constraints, attractors, and repellants — each exerting a type of gravitational force that deformed both the centerline of the path and its paving pattern (Figure 1.7). Attractors included elements such as entrances, landmarks and buildings; while repellants included existing trees, planned vegetation and certain land-uses. Moreover, the model incorporated elements ancillary to path placement, such as topography and subgrade infrastructure, in order to facilitate the collaboration between offices on a single software platform and 3D model.

35 Fletcher, David. "The Parametric Park." In Codify: Parametric and Computational Design in Landscape Architecture. Edited by Bradley Cantrell and Adam Mekies. (New York, NY: Routledge, 2018): 71-76.



a//Path generation process



b//Finalized weighting factors

//Figure 1.7

(a) South Park design process of feature prioritization and weighting and (b) pathway generated response to weighting factors and programmed constraints. (Fletcher Studios, 2018. Image from Fletcher, “The Parametric Park,” 73.)

1.4//Project Significance

i. A Status Quo of Digital Disparity

The future of algorithmic modelling in landscape architecture has an immense potential. For years, landscape architects have explored how to design the unique medium of landscape. Landscapes are enormously complex and complicated—presenting designers with an extensive palette of built elements and natural systems. The overall advancement of computational design in recent decades promises a means of abstracting these complex systems into something that is useful in the hands of designers. This is a matter of research as well as practical application in the landscape architecture profession. The current generation of landscape architects is emerging within a significant expansion of computational design tools and techniques.[36]

Simultaneously, a great many digital tools with sophisticated simulation functions are continually being developed with a decidedly architectural and engineering focus—such as the office standard Revit.[37] In turn, these specialized tools are adopted by landscape architects—who then must settle for the tools' limited applicability to their own design process. This should not imply that landscape architects have been slow to adopt cutting edge techniques and technology. The examples from Section 1.4 prove otherwise. Rather, this is largely a result of the relationship between the design fields and the business of technology—specifically that of software development. Put simply, architecture and engineering make up a considerably larger customer base than landscape architecture, therefore developers are incentivized to cater their software to those fields' conventions.[38] The commercial tools available to landscape architecture are comparatively generic as they are derivative of an architecture toolset. They fail to provide us with landscape design generating capabilities and are commonly relegated to the rendering of designs otherwise conceived by conventional means.

Meanwhile, our engagement with landscapes in the field has taken a digital turn, with advancing methods of remote sensing, an abundance of user data and an increasing frequency of augmented reality visualizations.[39] In this we have the potential to customize design solutions to real world conditions with greater alacrity, but only through the use of complementary design tools. In order to make up for the technological shortfall, digital design tools need to come from within the discipline.

Of the countless potential applications of computational design to landscape architecture, one that I believed to be ripe for development was stormwater management at a site-level. When considered as one of many aspects of site design, stormwater solutions are often initially conceptualized by landscape architects using "rules

36 Cantrell & Mekies. "Coding Landscapes," 6.

37 Belesky. "The Green Grasshopper," 406.

38 Ibid, 406.

39 Cantrell & Mekies. "Coding Landscape," 21.

of thumb" or simplified sizing guidelines. Details then emerge from the back-and-forth communication between the landscape architect and civil engineers. This process would be vastly improved if initial concepts were aligned with localized hydrological conditions and performance targets.

Stormwater simulation software does exist for this purpose. However—in keeping with supply-and-demand dilemma described before—that software is geared towards other disciplines. For instance, the EPA Storm Water Management Model (SWMM) software engine is used for a suite of sophisticated stormwater calculators that are proficient at numeric simulation but have no design capabilities. Furthermore, it is designed to simulate regional and watershed scale hydrological scenarios, not the small sites that are typically in the purview of landscape architecture. Civil 3D is another software option that is perhaps a step up in terms of design capabilities. Unfortunately, it seems an unlikely fit for landscape architecture designs because of its imposing depth in technical details. In order to aid landscape architects, a more versatile tool would be preferable.

ii. Algorithmic Tools for Landscape Solutions

The motivation to develop custom design tools fits into a greater trend of the creation and sharing of user-developed custom scripting. This trend is especially pronounced within the community of Rhino+Grasshopper users, a community which spans a wide variety of different disciplines and knowledge levels. The size and success of this community is due to the lower interface barrier of Grasshopper's graphical user interface (GUI) platform[40]—a feature that welcomes users of all backgrounds and skill-levels. Additionally, the sharing of user-made scripts is free and open through online portals. This trend is categorically different from the historically commercial approach to design software development, which is inherently closed and centralized.

The application of Rhino+Grasshopper to the field of landscape architecture is being explored through the efforts of a landscape architectural user community. Since Grasshoppers's inception in 2007, custom scripts have been gradually accruing, expanding on landscape-focused functions. Within the programming community, there is the notion of a *stack*, or rather, the functionally-linked layers of computer systems and software that collectively form the foundation from which new work is added. Successive layers of tools become the stacks that then form the foundation of increasingly specialized new developments. More conceptual than technical, this makes for an apt analogy for the utility of this project. A major part of the original contribution of this project is the direct, incremental addition to the landscape architecture stack. The efficacy of this project relies heavily on the additions that have made up the stack in its current form and, if successful, could lead to further advances and refinements in the future.

40 The distinctive innovation of Grasshopper is its visual programming language—meaning it is coded graphically and diagrammatically instead of with lines of text. Compared with text-based programming languages, Grasshopper is easy to learn and master.

1.5//Project Goals and Scope

This project sought to find a new way in which computational design might be employed in landscape architecture—using algorithmic modelling for stormwater design as a means of demonstrating that potential. Specific to that scenario, the goal was to explore the means of generating design solutions for site-scale stormwater management systems. The ultimate deliverable of that effort would be a functional Grasshopper algorithm, capable of generating custom stormwater solutions for unique site conditions.

This approach potentially offers a wide range of novel capabilities to the design process—not all of which could be adequately plumbed in the timeframe of this project. Through the course of background research, I gauged that the most worthwhile aspect of this approach is the capacity for semi-automated *solution optimization*[41]—the algorithmic merchanism for finding near-optimal solutions for complex problems. Several of the precedent studies demonstrated how designers were able to algorithmically ascertain optimal spatial and formal configurations of a variety of different landscape elements.[42] In a slightly different vein, it was discovered that a multitude of stormwater simulators provide precise hydrological analysis. Coupled with existing standardized methodologies for stormwater management planning, digital simulations offer the potential to devise an approach in line with computational design. The opportunity seized by this project was to use these standardized analytical techniques to provide performance parameters to a solution optimization process with spatial definitions—such as those conducted in the precedent studies. Efforts to this end were framed by the following research question and attendant goals:

> *Can a computational design approach be used to algorithmically generate site-scale stormwater solutions that are optimized to site constraints and performance targets?*

Goal 1—Stormwater simulation: characterize hydrological conditions on the site in response to design storm parameters

Goal 2—Generic solution: develop a semi-automated process for siting, sizing and shaping stormwater facilities.

Goal 3—Optimized solution: develop the algorithmic process to customize solutions to site conditions and performance targets.

Taken on their own, any results of the algorithmic stormwater solutions have unsubstantial value as knowledge gained.

41 *Solution optimization* is an algorithmic process in which a best-fit, near-optimal solution is calculated in response to a problem as it is formulated by the programmer. The term "solution" is somewhat problematic as its regular usage implies a sense of correctness. However, in this usage the solution is merely the outcome which is automatically generated within the predefined parameters and constraints.

42 Bassolino & Ambrosini (2016); Fletcher (2018); Wallis & Rahmann (2016).

That is because digital models work within a set of conceptual limitations. While they are progressively becoming ever more in-depth and multivariate, models will always remain abstractions of reality. Researchers working in the natural sciences use models in a very conservative way, operating under the assumption that every model is essentially wrong, but may still yield some useful data.[43] I am making a similar assumption and recognizing that this approach may yield good results that are only part of the overall picture of site design. That sentiment guides the second research question:

> *Can the design solutions generated by this approach be feasibly incorporated into a real-world site design scenario?*

Goal 1—Design options: generate multiple design options, beyond what is typical to conventional design process.

Goal 2—Cost-effectiveness: find the most efficient solution in terms of cost versus performance in runoff reduction and filtration.

Goal 3—Validity of reuslts: any generated solutions need to be verifiable independent of the algorithm itself.

As a means of addressing this research question, I surmised that the results of the computational design approach should be evaluated against a professional design—one that accounts for the breadth of typical design considerations and is representative of an expert-level of landscape architecture knowledge. Only through an objective comparison could this essential question of validity be answered.

43 Felson, Alex. "The Role of Models to Inform Green Infrastructure." (Presentation at Simulating Natures Symposium, Philadelphia, PA, March 2015. Accessed Jan 2019. https://www.design.upenn.edu/landscape-architecture/events/simulating-natures)

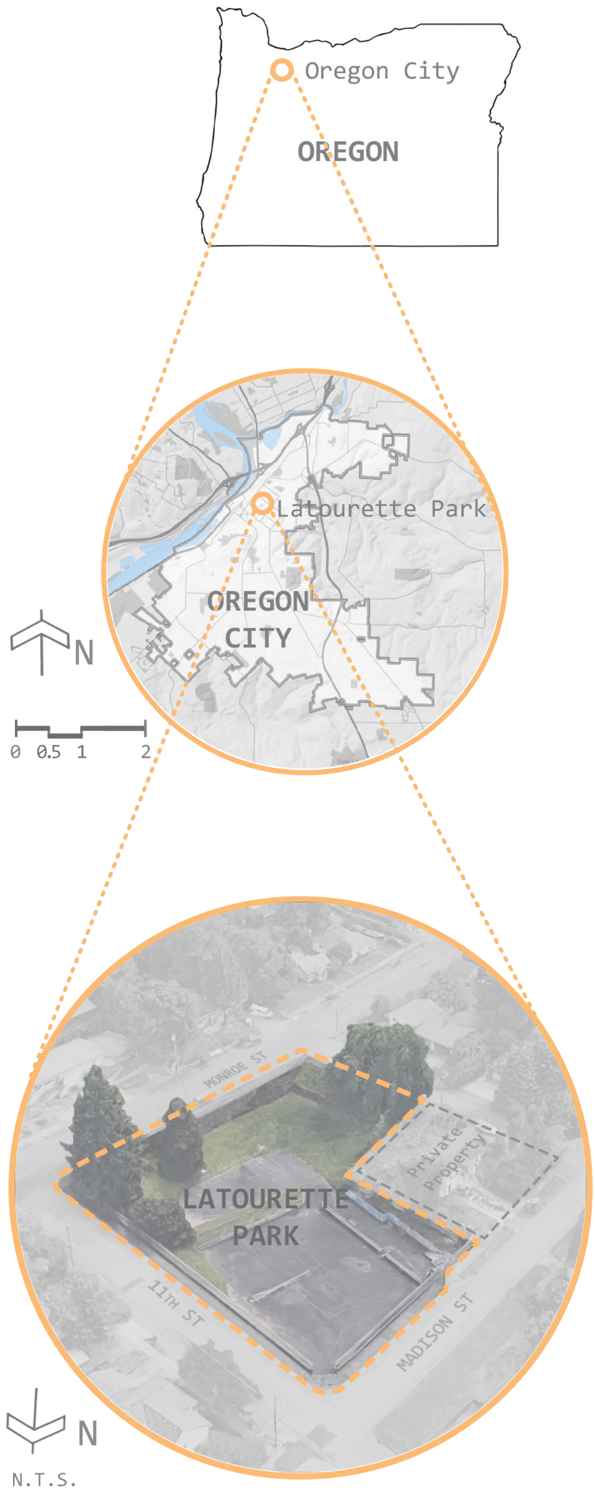
1.6//Site Introduction

During the course of research, the ideal project location arose with Latourette Park in Oregon City, OR. I was personally aware of the site through participation in neighborhood-led efforts in revitalizing the space, and came to learn that new designs were being drafted by the respected Portland-based landscape architecture firm GreenWorks. Founded in 1987, GreenWorks has assembled an extensive catalog of design work and natural resource management services across the Pacific Northwest. They have accrued approximately 100 awards for projects ranging from site design to restoration planning.

Most important in regards to this project, GreenWorks offered its cooperation as well as permission to refer to its documented design work for Latourette Park. As of the time that this project was executed, the GreenWorks design had passed the 30% construction document set milestone and was progressing towards a 60% set. The former would serve as a basis of comparison for project results, and provide accurate spatial data for site modeling.

i. Site Description

Latourette Park began as a public space in 1936 a public pool in the McLoughlin District neighborhood of Oregon City, OR (Figure 1.8). McLoughlin District is a streetcar-era neighborhood, composed mostly of early 20th century single-family homes in a grid of standard 200-ft by 200-ft blocks. The 0.8 acre park occupies roughly three-quarters of one of these blocks. It has the rather atypical quality of being cut



//Figure 1.8
Context map of Latourette Park within Oregon City, OR. (Base map imagery from Google Earth Pro)

a//North entrance

b//South wall

c//Former pool platform

d//Park tiers

//Figure 1.9

Existing conditions in Latourette Park, Oregon City.

into a hillside, so that all sides of the property enclose the interior like a quarry pit. Its single entrance at the north corner is the only spot at grade with the perimeter sidewalk (Figure 1.9.a)—the rest of which is separated from the interior space by fencing, side slopes and retaining walls. An imposing 15-ft wall of mortared fieldstone makes up the southeast edge, which is met at the bottom by a steeply sloped lawn (Figure 1.9.b).

Unfortunately, long-term deferred maintenance of the pool facility ultimately led to the decision to fill it and pave the site over in 1969.[44] Of the former pool, a concrete platform remains on the southwest edge (Figure 1.9.c) while the rest of the space was covered with asphalt.

Currently, Latourette Park is fairly deteriorated and chronically underused. Generally, its surface is divided into two relatively level tiers with a rise of approximately four feet between them (Figure 1.9.d). On the bottom level, basketball hoops and a tennis net were added some years ago but, like the pool, fell into disrepair.

ii. Community Context

Citizen involvement in the park's transformation started in 2016 with local Girl Scout Troop 4506 engaging local residents over the future use of the space. Coordinating with the Oregon City Parks Advisory Board, the movement gained public support and momentum. By summer 2018, significant progress was made towards realizing

44 "D.C. Latourette Park," (Oregon City website. Accessed October 2018. http://www.orcity.org/parksandrecreation/Latourette-park).

the park renovation. Under the direction of the non-profit organization DePave, neighbors gathered together in June to tear up and remove approximately 3,500 ft^2 of asphalt from the upper terrace (Figure 1.10). The goal of the DePave event—beyond fostering community involvement—was to clear the ground for future improvements.

Also in the summer of 2018, Oregon City Parks and Recreation had slated the park for early phase improvements in the 2020-21 biennial budget and produced a basic concept plan that reflected the collected opinions of local residents. The plan—prepared by Oregon City Parks and Recreation director Phil Lewis—envisioned the space as a much improved version of the current layout, with a resurfaced sports court, a nature play area on the terrace level, a pollinator garden and added vegetation (Figure 1.11). At this time, GreenWorks was commissioned to further the city's plan and design work proceeded under the project management of firm associate, Ben Johnson.

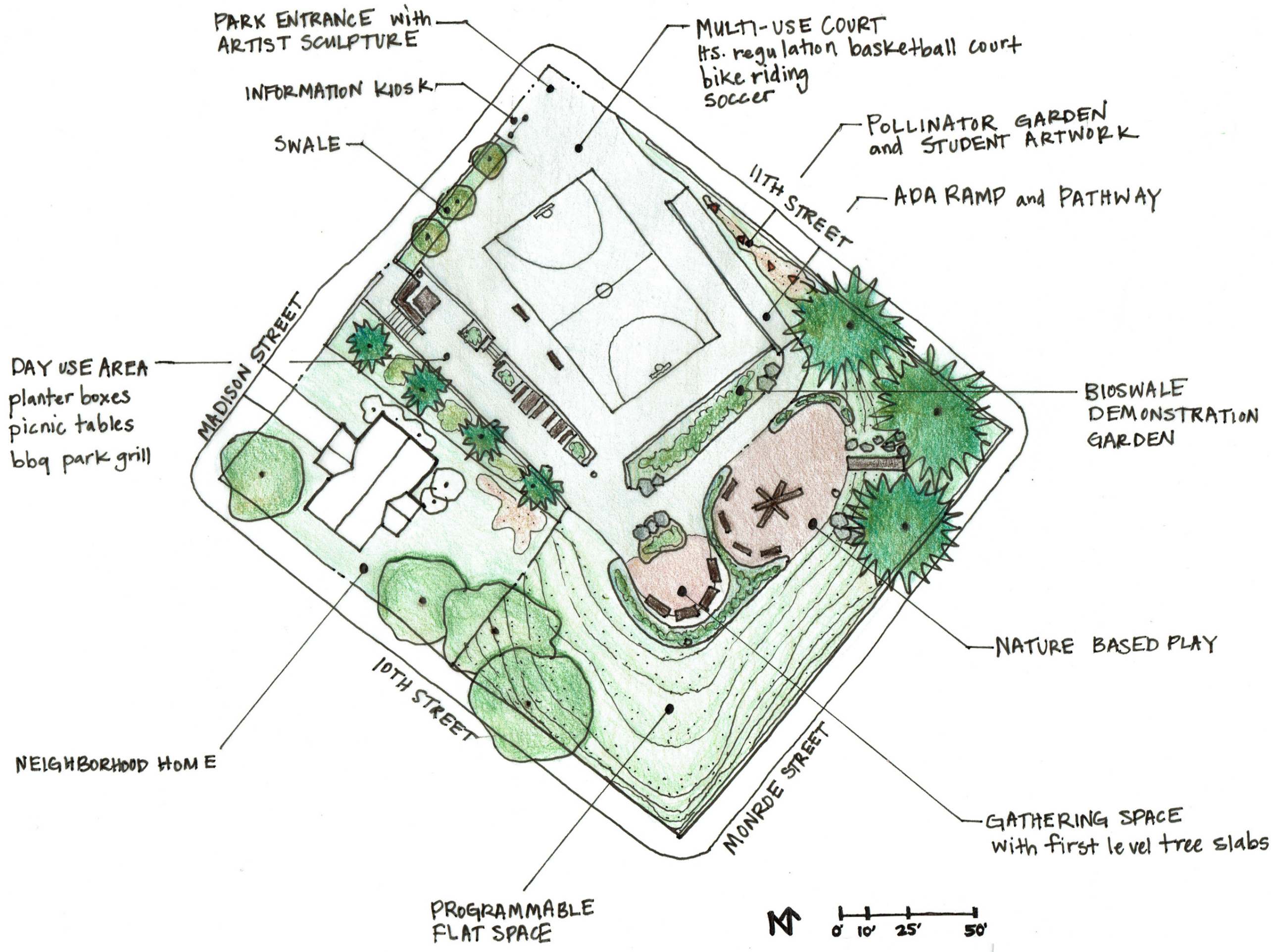
iii. Reasons for Selection

Hydrology

Due to the way the site has been cut into the hillside, it functions as one large basin with retaining edges and a slight slope towards the north end. The steady slope results in a prominent surface flow direction which simplifies the onsite hydrology despite minor depressions where drainage collects. Simultaneously, the raised curb of the perimeter sidewalk ensures that there is a negligible amount of surface flow entering the site from outside of its bound-

//Figure 1.10

DePave event at Latourette, summer 2018. (photographer unknown, Facebook-DC Latourette City Park, 2018)



//Figure 1.11

Original concept plan for improvements to Latourette Park put forth by Oregon City Parks and Recreation. (Graphic credited to Phil Lewis, https://www.orcity.org/sites/default/files/Latourette_park_concept_map_2018.pdf)

ary. Typically, site analyses of hydrology are complicated by a catchment extent that goes beyond the site's property lines—here that complication is avoided.

Subjectivity

The redesign of the park presented an opportunity to fit into the broader project framework and circumvent the shortcomings typical of the modeling strategy. It is obvious that the modeling strategy is thoroughly suited to weigh quantifiable factors. However, it is not the best means of interpreting intangible, unquantifiable aspects of design. These would be important sociocultural considerations such as stakeholders' attitudes, goals, and vision for the space. As it happens, the redesign of this park space has been a community-led endeavor which has succeeded in promoting the community's collective voice. It was therefore an assumption that the city's concept plan had been vetted for

“There is a tension between the productive abstraction of diagrams and the need to rigorously resolve their assertions against the actual conditions of a site. The computational design process offers a means to release this tension.”

—Philip Belesky[1]

1 Belesky, Philip.“Adapting Computation to Adapting Landscapes,” *Kerb*, no. 21 (2013): 54

2 Methodology

2.1//Project Methodology Overview

i. Strategy of Inquiry

The upcoming chapter summarizes the stage of this project that pertained to gathering and synthesizing knowledge. In keeping with academic protocol, the project research methodology is presented within the context of research in landscape architecture and is shown to align with established modeling strategies. Separately, the complete project methodology is sequenced as a lead-in to the design stage of the project, which follows in Chapter 3.

The overarching approach of this project was in developing a generative landscape design guided by multi-disciplinary knowledge. As an essential concept, the term *generative* implies a meta-design process—one in which the design emerges systematically, and indirectly, from the actions of the designer. In order for a generative design to function, it needs to be governed by explicit rules and relationships—as opposed to subjective inferences. Therefore, the broad methodological approach of this project is shaped by an objectivist research strategy. Objectivism refers to the type of approach that has some inherent means of validation which is independent of the researcher's influence. It generally pertains to research that uses empirical data and quantifiable relationships. Much of this project concerns the biophysical processes that occur within landscapes. These processes are indeed quantifiable, but also have an imposing tendency for organized complexity.[2]

2 Refer to Section 1.2.i for context of how *organized complexity* relates to design theory.

Studying such complexity can be aided with a modeling strategy. There are many types of models that are geared for different types of inquiry, but a commonality with all models is simplification through the abstraction of real-world conditions and the incorporation of selected empirical data.[3] The vehicle for this project is an algorithmic model developed with Rhino+Grasshopper.

Rhino+Grasshopper has qualities that are somewhat unique to the modeling strategy typologies. It is a digital tool that has an interface split between two modes—direct manipulation in 3D model space and model encoding via a visual programming language. In this way, the resultant model straddles the classification distinction between modeling for external representation or internal representation.[4] It simultaneously provides an interactive visual representation and accounts for the dynamic behavior of elements/systems in the landscape. As such, this modeling technique exemplifies dynamic simulations, which is useful for studying changes in the landscape. The particular real-world conditions abstracted in this project were site-level hydrology, and stormwater interventions that were empirically evaluated through encoded mathematical functions.

This project examines the interception of two areas of academic/practical knowledge that work off relatively disparate frameworks. Stormwater management uses hydrology simulation and decision models as a method of ordering and quantifying landscape systems for design responses. Computational design is more generalizable but is typically involves a high degree of abstraction as a means of optimization between form and function. Its operational framework is ordered through process models. Despite the fundamentally different subject matter between the two frameworks, they have certain structural similarities. My aim was to use that structural similarity in order to hybridize the two—primarily retaining the logical operation of the computational design framework which is also informed by the stormwater logical operation. Doing so was a matter of drawing out the various framework components from stormwater and correlating them to the framework components of computational design. A new framework was developed to direct the algorithmic model development—as detailed in Section 2.2.

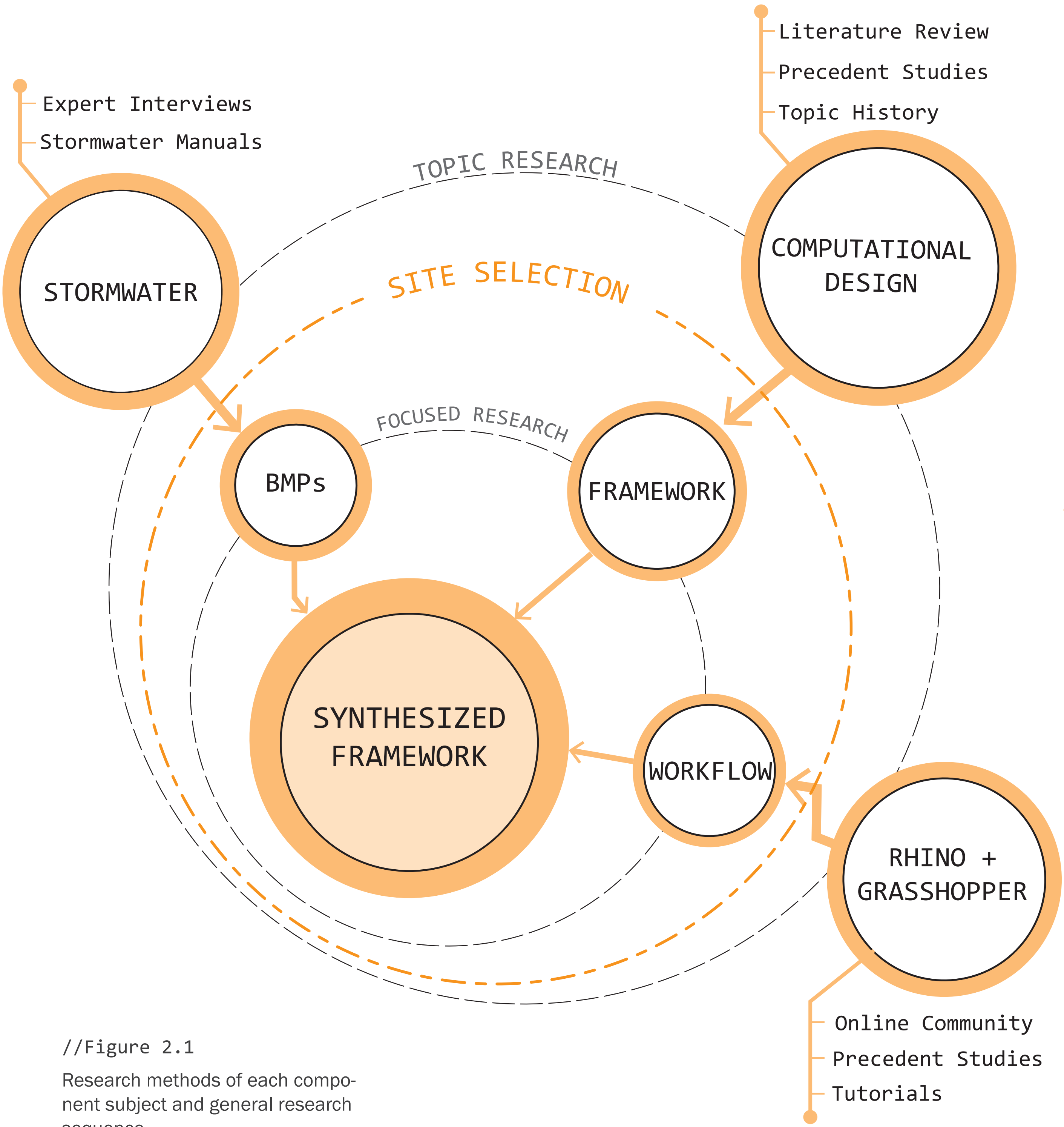
ii. Research Methods

Various research methods were used in the lead up to the development of the algorithmic model. Altogether, the project required the research of three topics: stormwater management, computational design and the use of Rhino+Grasshopper. Research methods varied because of the nature of information associated with either topic. Figure 2.1 outlines those research methods which coalesce at the creation of the synthesized framework.

Stormwater management is largely a practical matter of applied knowledge. Its methodologies are codified into industry standards and government regulation. For that reason, information on this topic tends to be procedural. Primary information sources for stormwater included tech-

3 Deming, Ellen & Swaffield, Simon. *Landscape Architecture Research: Inquiry, Strategy, Design*. (Hoboken, New Jersey: John Wiley & Sons Inc., 2011): 87.

4 Ibid, 87.



//Figure 2.1

Research methods of each component subject and general research sequence.

nical hydrology manuals and governmental code—both of which are detailed in the next chapter. Additionally, expert knowledge was sought for stormwater management design and to characterize the study site. I conducted informal interviews with GreenWorks landscape architect Ben Johnson and Oregon City Parks and Recreation Director Phil Lewis—also of a landscape architecture background. Both individuals were able to provide background knowledge as well as design documentation of the study site.

Computational design is the body of theory underpinning the modeling strategy. For this type of knowledge, I used literature review as a research method—referring mainly to Cagan *et al*. "Computational Design Synthesis Framework" (2005).

The approach to researching Grasshopper was altogether different. The knowledge is procedural in nature but it is more skill-based than it is referential. At the onset of this project, I had little experience with Grasshopper. And while it is accessible to new users relative to other programming methods, the concepts and structure of the programming language were considerably complex. Learning this language involved a process of self-education using sources such as tutorials, precedent analyses and correspondence with the online community of Grasshopper users.

The following sections of this chapter detail key aspects of the project methodology.

2.2//Synthesis Framework

The following section summarizes the process in which the synthesized framework for the algorithmic modeling approach was developed. It first presents the selected component frameworks for computational design and stormwater management design, as well as a concise justification for their selection. It then depicts the logical basis of combining the two, based upon the structural and conceptual similarities.

i.Computational Design Framework Overview

Using the approach of computational design takes a landscape design scenario and fundamentally changes both the design medium and the design environment. The computational design process is not about directly producing solutions for the design scenario. Rather, the approach is about designing a search process[5] for a multitude of potential solutions. It is a meta-design process, and it repositions the designer away from the familiar territory of conventional design and into the digital design environment.

To guide this effort, I will be referring to an established framework for computational design developed by authors Jonathan Cagan, Matthew Campbell, Susan Finger & Tetsuo Tomiyama in their article "A Framework for Computational Design Synthesis: Model and Applications." By the authors' own definition, computational design is "...the algorithmic creation of designs; the organized, methodological modeling, implementation and execution of design creation on a computer."[6] Ideally, this framework is applied to problems in which the designer does not have the option of a formulaic method to an "ideal" solution, but instead may require a multitude of alternatives to compare.[7] It is a broad framework for solution optimization—which is essentially an algorithmically driven process of elimination executed through a variety of methods.[8] This particular framework was selected on the preconception that solving a landscape architecture "problem"—which is intrinsically multivariate and spatial—would necessitate a search method that is geared toward the indeterminant.

As a point of reference, there are many parallels to be drawn between this framework for and the general methodology of any conventional design process. Any application of this framework involves a cyclical execution of the following stages[9]: *representation*[10]*, generation, evaluation*

5 *Search process* refers to the systematic method that the algorithm uses to generate and evaluate solutions.

6 Cagan et al., 171.

7 Ibid, 178.

8 There are several algorithmic search methods that may be broadly classified under solution optimization. Each employs a different machine learning strategy to progressively improve search results—enabling them to solve complex problems with relative efficiency. Those methods include but are not limited to knowledge-based searches, genetic algorithms, agent-based searches and shape grammars. Ibid, 175-178.

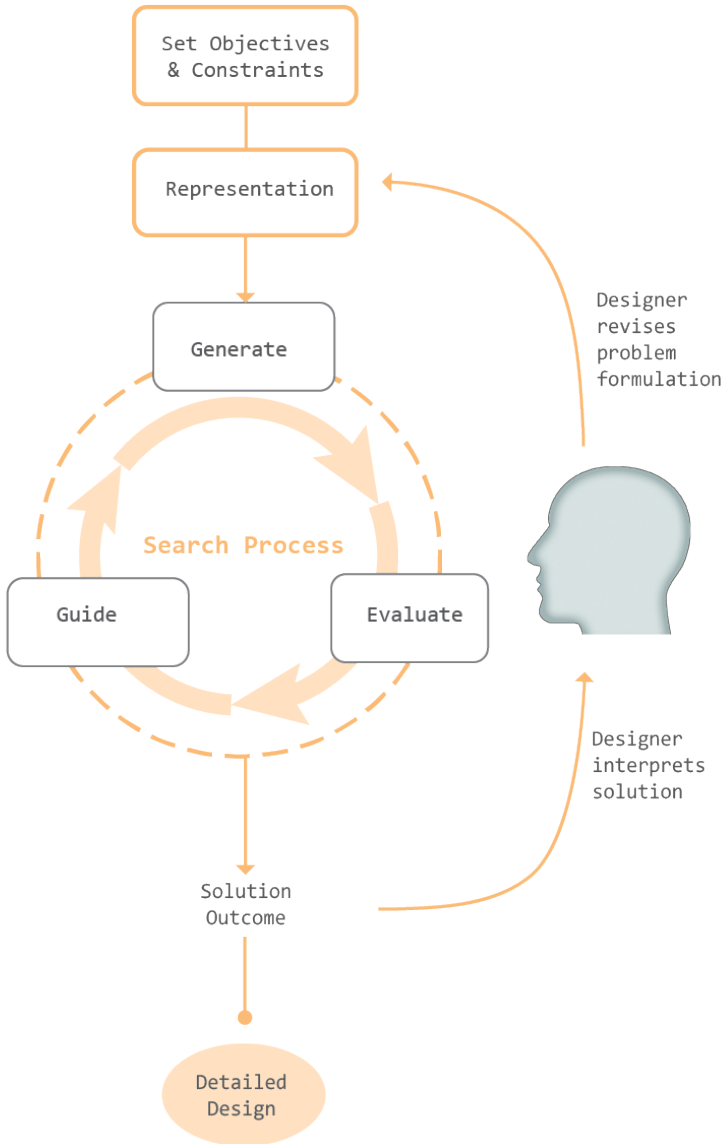
9 Ibid, 171.

10 Unless otherwise noted, the term *representation* is in reference to its usage in computational design, and not a shorthand for 'visual-representation' as is typical in design fields.

and *guidance* (Figure 2.2). The first step, representation, is akin to the conventional design step of problematizing the design scenario and creating a mental model. Generation is simply the creation of component parts and their subsequent assembly into the whole—similar in concept to drafting different design alternatives. Evaluation, as it would seem, involves analysis of how well the previously generated outcome meets performance goals. And lastly, guidance is the means of feedback in order to improve the next generated outcome. The stages of generation, evaluation and guidance automatically cycle until the generated outcome can no longer be improved. Based upon the quality of the final outcome, the designer may then return to the representation stage and remodel the entire problem as needed. So while the computational design environment may be foreign, its concepts are entirely intuitive—being similar to any design process of gradually improving designs through iterations. The key difference with this approach is that design improvements occur as minute changes across numerous iterations, but through a largely automated process. Each of the fundamental framework stages is discussed in greater detail in the following section.

Representation

A key concept of this design framework is that the complexity of the algorithm—and consequently the comprehensiveness of the solution(s)—is entirely controlled by the designer within the representation stage. Representation is a process of translation wherein the problem scenario is converted into objects and attributes recognizable within the computational



//Figure 2.2

Computational Design Framework. Adapted from Cagan et al. "A Framework for Computational Design Synthesis," 172.

design environment. Most crucially, this involves defining what the constraints and objectives are and then relating them in quantifiable terms to the search process. Having a higher number of constraints and objectives—the basic building blocks of the problem—results in a wider range of solution outcomes and a proportionally slower search process. On the other hand, having relatively few building blocks makes for a simpler, easier search process; but also one that may exclude all possible solutions. In this sense, the problem formulation of this stage is just as, if not more, important than the problem solving of the subsequent stages. For this design project, the translation of the problem involves manipulating geometric data within a virtual space. This data represents real-world phenomena and the programmed responses to the resultant constraints. Like the engineering applications that this framework was devised for, the problem scenario of this project is one of searching for a "form" that meets prescribed functional requirements. Although it differs is that the problem incorporates spatial context and abstractions of fluctuating natural systems.

Generation

Working within the constraint parameters established in the representation step, the generation task automatically produces candidate solutions using any number of methods. These methods may be as naïve[11] as a random number generator or based on a more sophisticated search procedure. Whatever the method may be, this step occurs across multiple iterations, with each generated solution cycling through the subsequent steps of evaluation and guidance.

Evaluation

Evaluation is the first part of a feedback process that is used to progressively refine solution outcomes. As one would assume, the evaluation step tests the potential value of each of the generated solutions. It may be accomplished directly, through a mathematical expression, or involve multiple iterations of a complex simulation. The process may be further complicated by having multiple objectives to test. Often that situation is simplified by converting multiple objectives into a single objective — which is far easier to solve.

Guidance

Guidance is the second part of the feedback process. It responds to the values determined in the previous step and adjusts the search process accordingly. Typically, guidance is seeking a maximum or minimum value for the objective(s) through an optimization process. For instance, when solving for monetary cost, it would be more favorable to have the lowest value in the solution. If—in a sequence of solution iterations—a solution was measured to have a higher cost than the previous one, then the guidance process would reject the parameters that produced the higher cost. Logically, it would then direct the subsequent solution to reverse the direction of changes to parameters, ostensibly lowering the value for cost.

In the case of this project, the guidance will involve changing the geometric parameters of a shape within a solution space. Early iterations of the search process will

11 *Naive* in this context refers to an algorithmic routine that has a simple logic, such as searching alphabetically.

likely see a high variability in those changing parameters. But as the search narrows, that variability will decrease as changes in shape become more minute.

ii. Selecting a Stormwater Reference Standard

Designing for stormwater management is a process that has been significantly developed across multiple disciplines and is well-defined by established design standards. These standards vary with location and jurisdiction, often having been custom developed to meet individual challenges of local conditions. The regulation and rigor of a managment regime may also be a matter of incorporating different scales of design intervention into regional systems—such as when managing entire watersheds. Whatever the extent may be, stormwater design starts with defining performance goals based on location specific challenges and the values of the local community. With those ideas established, a set of best management practices (BMPs) may be codified as the standard for design, construction and management of stormwater infrastructure.

Two reference standards were used in the course of this project. *The Sustainable Drainage Systems* (SuDS) *Manual*[12], is referred to for hydrological calculations made in the project model. It also offers a framework via design guidelines and decision models for the selection of stormwater infrastructure components. This manual was selected both for its comprehensiveness and the broadness of its guidelines. Meant as a national guide, it does not limit its recommendations by local constraints and instead is applicable to a wide variety of hydrologic scenarios. Because of these qualities, it was reasoned that this project's outcome model would be more transferable to different locations and stormwater scenarios.

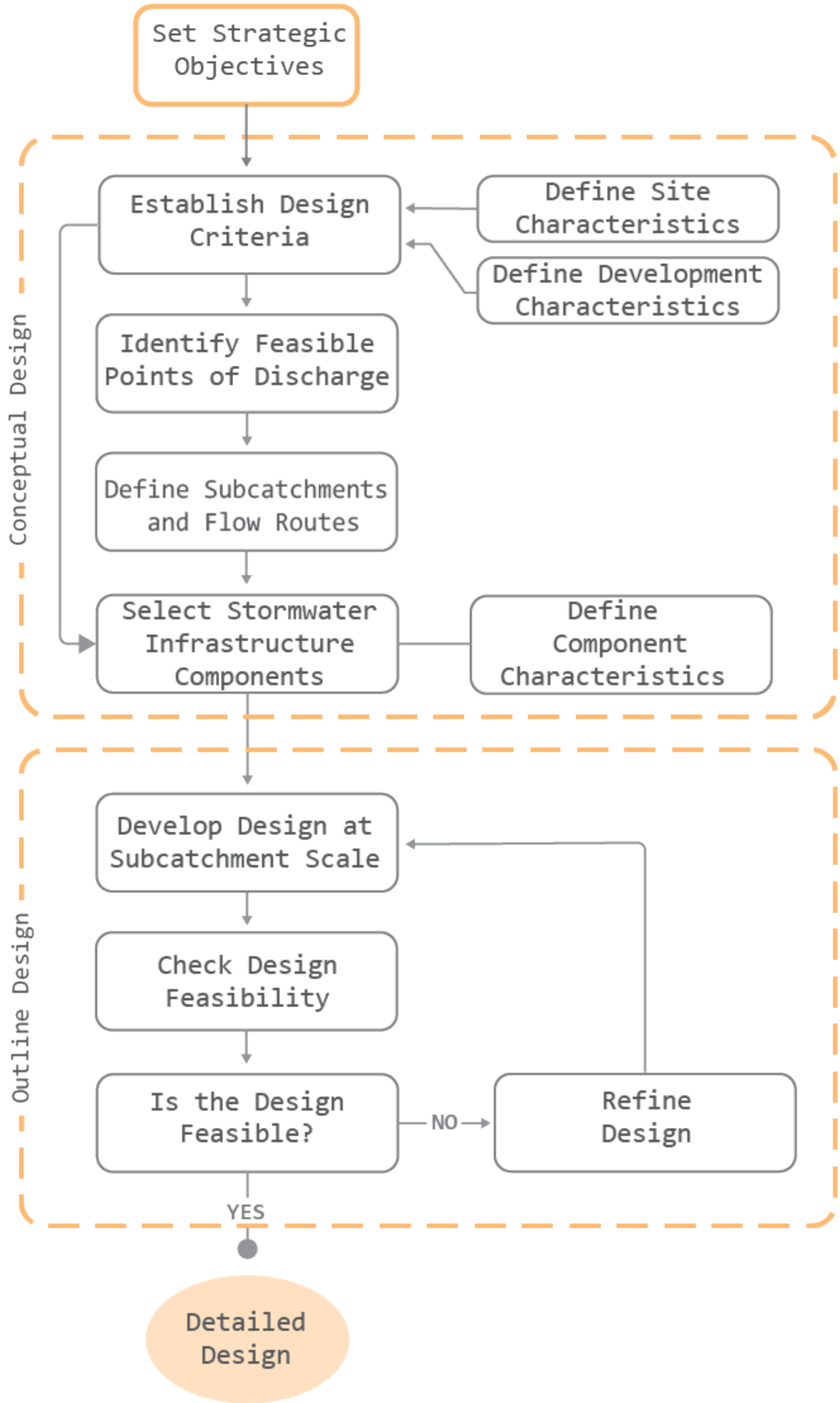
The *Portland Stormwater Management Manual*[13] was then referred to for design standards and performance goals for stormwater design at the Latourette site—as detailed in later sections.

iii. SuDS Stormwater Framework

The SuDS Manual gives extensive background on the stormwater design process, from early considerations of strategic objectives through to detailed design. The process is organized in a step-by-step guide that broadly follows four stages: strategic objectives, conceptual design (initial design and layout), outline design (sizing and optimization), and detailed design of the final scheme. The components of these stages are outlined in Figure 2.3.

12 Woods-Ballard, Bridget, et al., *The SuDS Manual*, (London: ciria, 2015).

13 City of Portland, *Stormwater Management Manual, 2016*. (Bureau of Environmental Services, Portland, Oregon, 2016).



//Figure 2.3

SuDS Manual decision model for site-based stormwater design.
(Adapted from Woods-Ballard et al, *SuDS Manual*, 101-119).

iv. Synthesized Framework

To guide the development, execution and evaluation of the algorithmic model, a custom framework was synthesized from the aforementioned frameworks of stormwater management and computational design. The two frameworks have a comparable structure and sequence, which was simplified as:

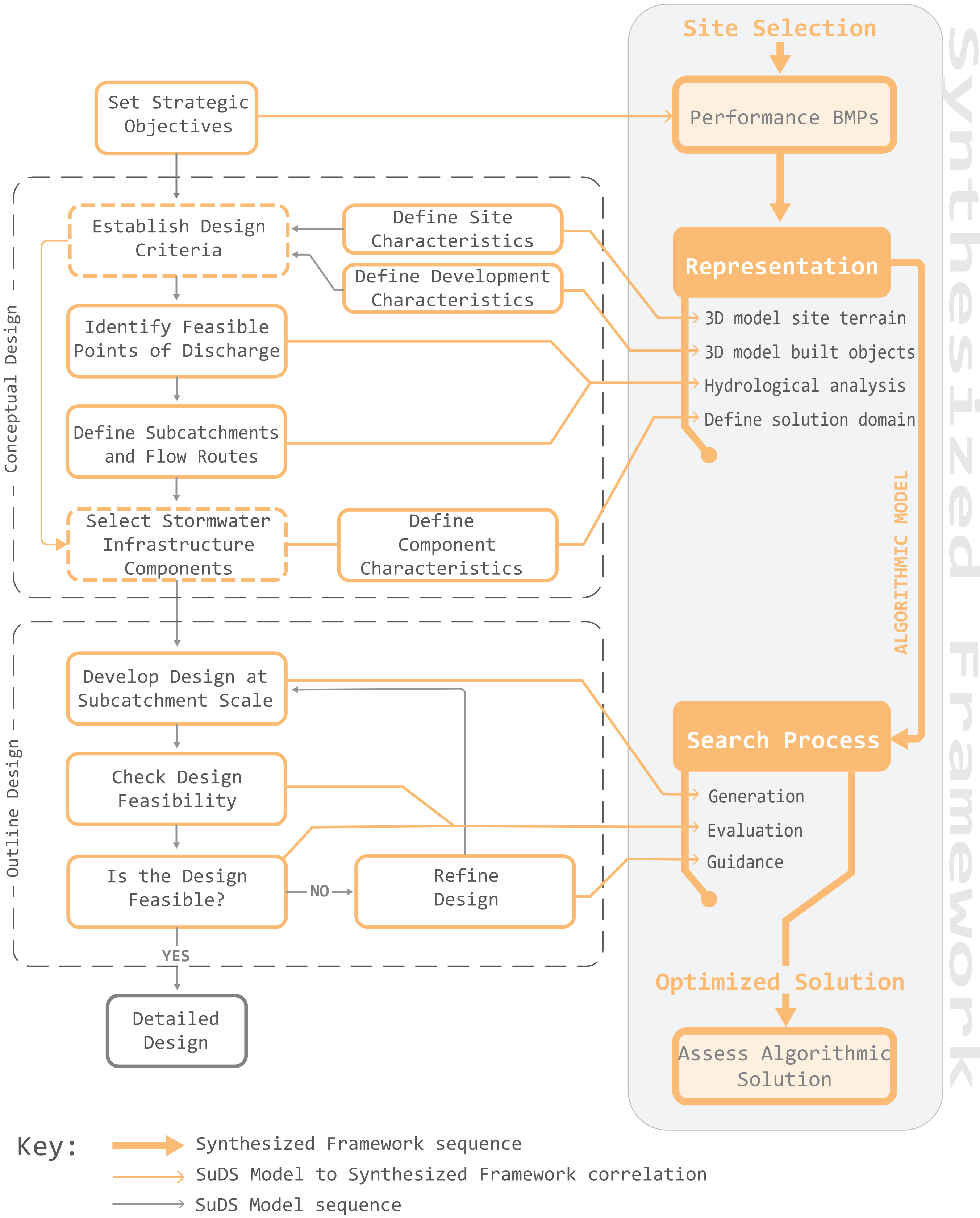
1) Start with a tangible definition of goals
2) Spatially characterize the problem, or site constraints
3) Develop a design for a specific component through a guided, cyclical process
4) Proceed to a detailed, overall design

Despite their broad similarity, an initial comparison of the two frameworks did not perfectly relate in a one-to-one fashion across these steps. It was also determined that, between the two, the Computational Design Framework was more consequential to the algorithmic model development. Therefore, emphasis was placed on preserving the essential ordering of its four key stages and the SuDS framework was slightly condensed. This preliminary step enabled a seamless synthesis in which direct correlations were drawn between the two frameworks (Figure 2.4).

The first step of defining goals and objectives was accomplished by simply referring to local stormwater BMPs—as detailed in the next section. Next, it was found that the two design stages of the SuDS framework neatly align with the stages of the Computational Design Framework. Specifically,

//Figure 2.4

Synthesized framework for the algorithmic modeling approach combining elements of the Computational Design Framework and *SuDS Manual* decision model.

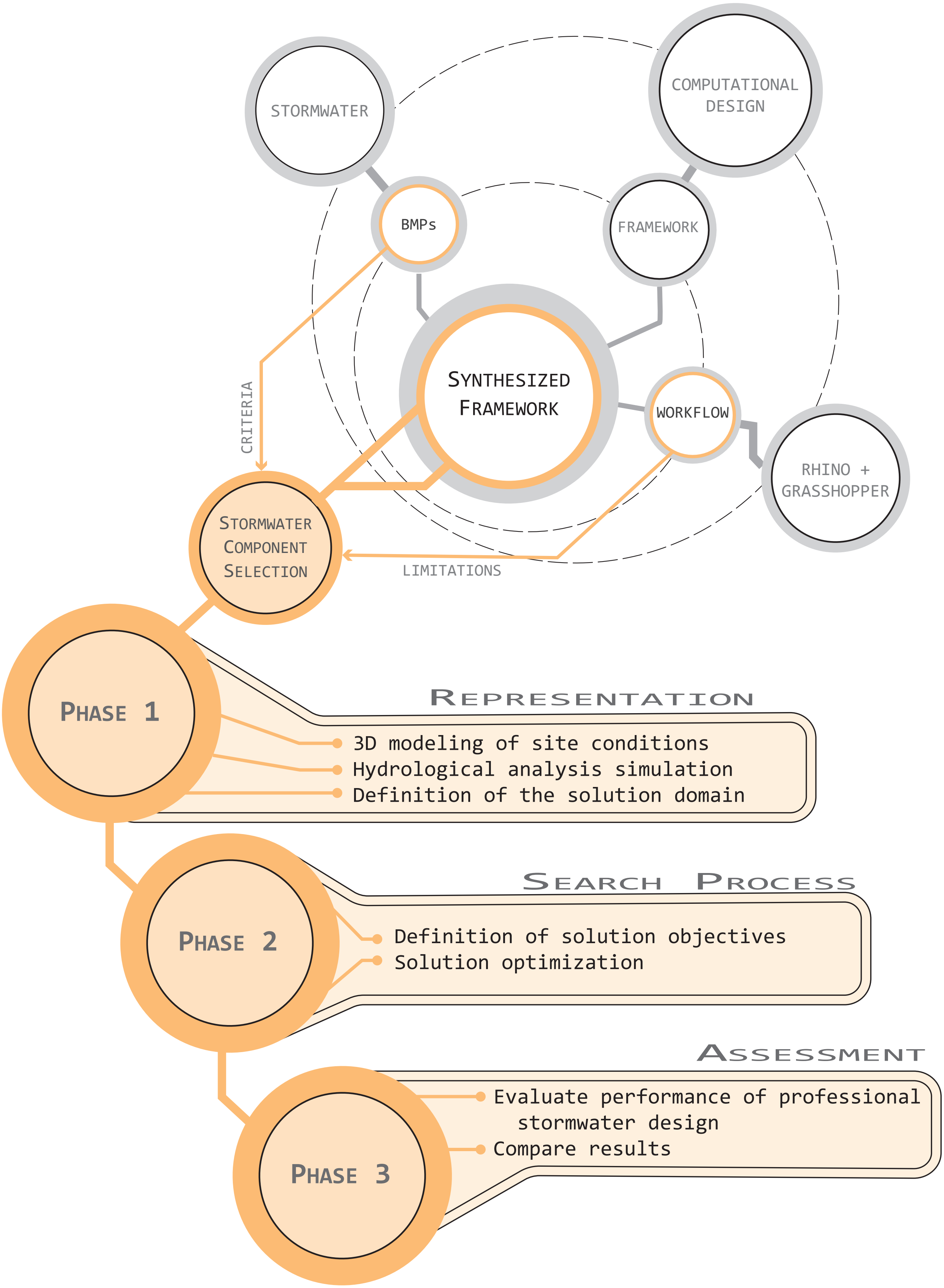

Site Selection
Performance BMPs
Representation
3D model site terrain
3D model built objects
Hydrological analysis
Define solution domain
ALGORITHMIC MODEL
Search Process
Generation
Evaluation
Guidance
Optimized Solution
Assess Algorithmic Solution
Synthesized Framework
Set Strategic Objectives
Conceptual Design
Establish Design Criteria
Define Site Characteristics
Define Development Characteristics
Identify Feasible Points of Discharge
Define Subcatchments and Flow Routes
Select Stormwater Infrastructure Components
Define Component Characteristics
Outline Design
Develop Design at Subcatchment Scale
Check Design Feasibility
Is the Design Feasible?
NO
Refine Design
YES
Detailed Design
Key:
Synthesized Framework sequence
SuDS Model to Synthesized Framework correlation
SuDS Model sequence

the ‘conceptual design’ stage systematically outlines the process of problem formulation and thus closely tracked with the Computational Design Framework stage of “representation.” Synthesis of the two involved translating the constraints of the design problem into information that would be usable within the algorithmic model. Next, it was determined that the ‘outline design’ stage of the SuDS model roughly corresponded to the Computational Design Framework stages of ‘generation,’ ‘evaluation’ and ‘guidance’—collectively the ‘search process’ that produces the algorithmic solution. The last step of the synthesized framework pertained to assessing the algorithmic solution. As abstractions of real systems, models require a means of validation. Therefore, this step outlined the process to substantiate the results of the model within a satisfactory range of accuracy through a means that was independent of the model itself. This involved the comparison of the algorithmic solution to the professional design for the same site and evaluating for stormwater performance metrics, cost-efficiency and subjective measures of feasibility.

Moving forward, the synthesized framework was integrated into the the project methodology as discrete phases to follow during the algorithmic design (Figure 2.5). The phases are primarily organized around the broad steps of the framework—so that Phases 1 and 2 pertain to the representation and search process stages , respectively. Phase 3 then follows the last frame

//Figure 2.5

Expanded project methodology which follows the algorithmic design section of the project.


STORMWATER
COMPUTATIONAL DESIGN
BMPs
FRAMEWORK
SYNTHESIZED FRAMEWORK
WORKFLOW
RHINO + GRASSHOPPER
CRITERIA
STORMWATER COMPONENT SELECTION
LIMITATIONS
PHASE 1
REPRESENTATION
3D modeling of site conditions
Hydrological analysis simulation
Definition of the solution domain
PHASE 2
SEARCH PROCESS
Definition of solution objectives
Solution optimization
PHASE 3
ASSESSMENT
Evaluate performance of professional stormwater design
Compare results

2.3//Stormwater Component Selection

This section describes the process in which the type of stormwater infrastructure component was chosen for Latourette Park. During my self-education in Grasshopper, I found that selecting between different infrastructure options within the model was a problem beyond my skill-proficiency to solve. Consequently, the algorithm was designed to only handle a single type of infrastructure. It should be noted that any process—conventional or algorithmic—for designing a stormwater management scheme would ideally accommodate a combination of infrastructure options. As such, this limitation was regarded as a setback. Technical limitations notwithstanding, a single, best-fit stormwater infrastructure component was selected based on the criteria detailed ahead.

i. Regulations and Best Management Practices

In consulting with the Latourette Park project manager at GreenWorks, Ben Johnson, I learned that Oregon City defers to the City of Portland for standards of stormwater management. GreenWorks routinely uses the published guidelines from Portland Bureau of Environmental Services (BES) for construction and performance standards of stormwater infrastructure.

Accordingly, this project refers to the 2016 edition of the *Portland Stormwater Management Manual* (PSMM) for technical requirements and performance goals in designing stormwater management schemes. Portland BES uses a simplified, overarching standard of performance for permitting stormwater infrastructure. Their goal is to maximize the mitigation of stormwater using a 10-year design storm[14] scenario with a goal for total onsite filtration/infiltration. That design storm is equivalent to 3.4 inches of rainfall over a 24-hour period. Stormwater is to be infiltrated onsite to the maximum feasible extent before discharging overflow. Other technical requirements pertain to dimensioning of stormwater facilities. Those are addressed in detail in later chapters.

ii. Selection Method

Per the original SuDS design guidelines, selecting the right type of stormwater intervention is a matter of aligning values-based goals with technical performance goals such as runoff reduction targets. The same process is carried over to the synthesized framework. Having established both the community's goals and the regulatory standards, it was possible to select the right stormwater infrastructure component from a comprehensive list obtained from the SuDS Manual (Table 2.1). Selection criteria—as gleaned from community and technical performance goals—included:

1] Amenities: biodiversity, attractive-ness, human comfort

2] Filtration/infiltration

3] High runoff reduction capacity

4] Low-cost

14 *Design storm* is a hydrologic concept that encapsulates the parameters of a particular storm event. Those parameters are used to calculate designed responses in stormwater management schemes.

Component Type	Description	Mitigation	Amenity Value	Constructability	Costs	Maintenance
Green Roofs	A planted soil layer is constructed on the roof of a building.	Filtration, attenuation ☑	Reduces heat-island effect ☐	Complex ☐	High ☐	Low ☑
Infiltration Systems	These collect and store runoff allowing it to infiltrate into the ground.	Filtration, infiltration ☑	Low ☐	May include constructed soakaway feature ☐	Low to Med ☑	Low ☑
Proprietary Treatment Systems	Complex structures that remove specific contaminants.	Filtration, adsorption ☑	None ☐	Complex ☐	High ☐	High ☐
Filter Strips	Grassed area to promote sedimentation and filtration of runoff.	Filtration, attenuation ☑	Low ☐	Basic ☑	Low ☑	Low ☑
Swales	Vegetated channels used to convey and treat runoff.	Filtration, attenuation ☑	Attractive vegetated corridor ☑	Basic but extensive footprint ☑	Medium ☐	Depends on extent of vegetation ☑
Bioretention Systems	A shallow depression allows runoff to pond before filtering through vegetation and underlying soils.	Filtration, infiltration ☑	Attractive vegetation, microclimatic cooling ☑	Basic ☑	Low to Med ☑	Depends on extent of vegetation ☑
Pervious Pavement	Runoff is allowed to soak through structural paving	Infiltration, attenuation ☑	Low ☐	Structural considerations ☐	Medium ☐	Med - High ☐
Attenuation Storage Tanks	Below-ground void spaces used to temporarily store runoff.	Attenuation ☐	None ☐	Complex ☐	High ☐	Low ☑
Detention Basins	Basic depressions that fill during heavy rain but are usually dry.	Attenuation ☐	Potentially adds biodiversity ☐	Basic but extensive footprint ☑	Medium ☐	Med - High ☐
Ponds and Wetlands	Permanent pool of water used to provide attenuation and treatment of runoff. Can support emergent and submerged vegetation.	Attenuation, filtration, biofiltration ☑	High biodiversity value and potentially attractive ☑	Complex ☐	High ☐	High ☐

//Table 2.1
Stormwater infrastructure component checklist. (Adapted from Woods-Ballard, et al., *The SuDS Manual*, 29.)

iii. Bioretention Facility Characteristics

Given the selection criteria, the best-fit stormwater infrastructure component for the Latourette Park site was decided as the bioretention facility.[15] This type of facility is known for its flexibility. It may be integrated into a variety of scales of stormwater management and may exhibit a wide variety in shape, size, filter materials and support of vegetation. Most commonly, they are used as a means of infiltrating and passively treating stormwater on-site. Typically, bioretention facilities are designed for total infiltration in frequent rain events, but are sized to allow inundation and spillover in rare, heavier storm events.

The general design characteristics for this facility are illustrated in Figure 2.6, and are as follows:

- Shallow depressions with level bottom and sloped sides at the surface level.
- Native vegetation selected for phytoremediation capacity and suitability to wet conditions.
- Constructed soils used as filter/growing medium which may include submerged anaerobic zones to promote nutrient removal.
- Drainage layer with optional underdrain of perforated pipe discharging away from the facility.

Unless the design scenario involves unavoidably high stormwater input velocities, then the construction of this facility is relatively simplistic—at most requiring overflow options such as dammed spillways and outflow drainpipes. Irrigation may also be required at installation but may be unnecessary once vegetation becomes established.

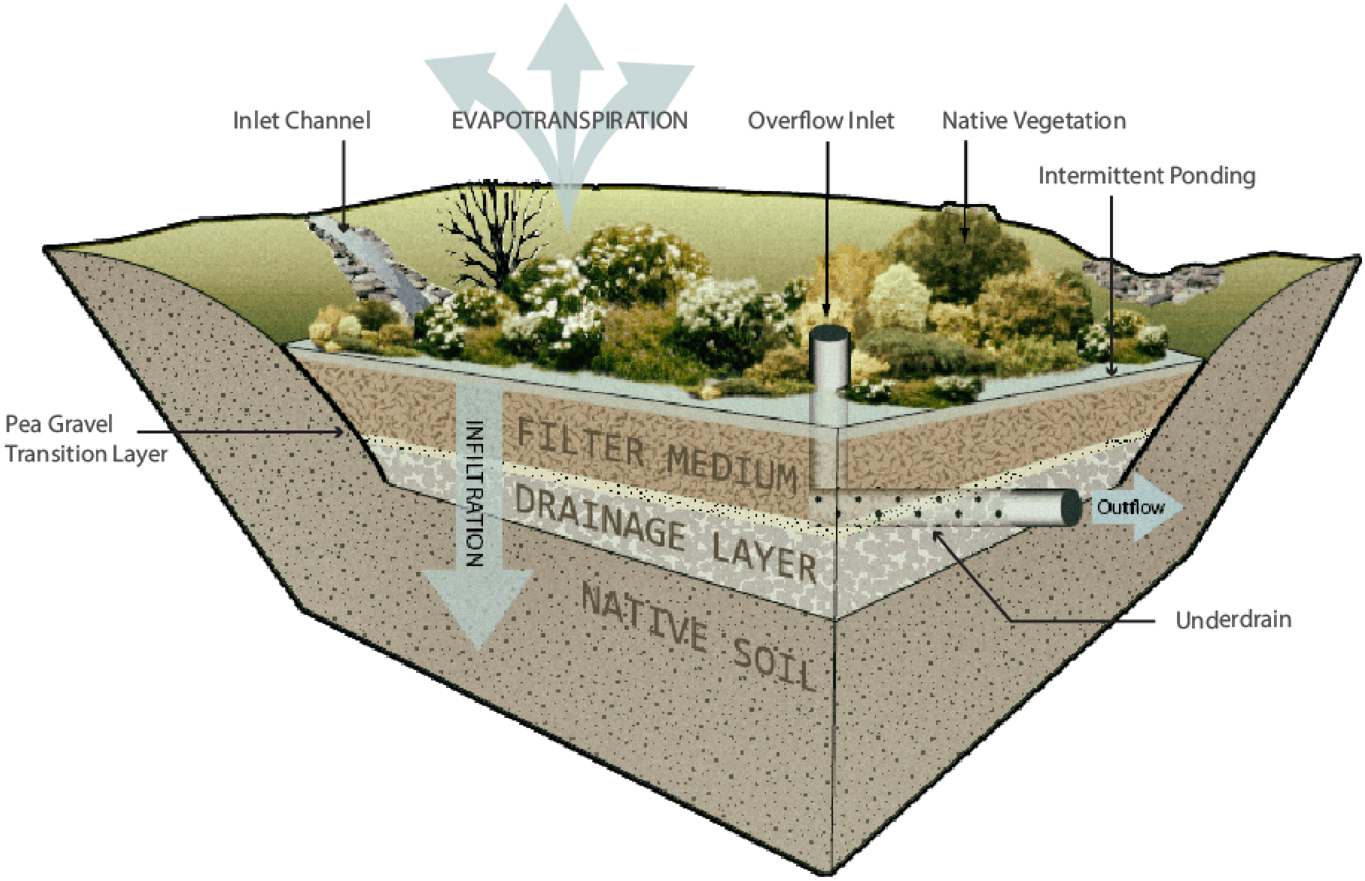
Under typical circumstances, bioretention facilities manage stormwater through several natural processes. Vegetation plays a major role in mitigating stormwater—first by intercepting rainfall before it collects on the ground and then through evapotranspiration. Root systems contribute with the uptake of nutrients and trace amounts of pollutants such as metals, organic compunds, fuels and solvents.[16] Additionally, plant roots increase both the infiltration rate and capacity of the soil. Infiltration and filtration are the primary means of processing stormwater. Compared to other types of stormwater infrastructure, it has a high mitigation index for the filtration of suspended solids, metals, and hydrocarbons.[17]

Aside from the stormwater functionality, bioretention facilities offer many other benefits. They are attractive, naturalistic landscape features that support biodiversity. Generally, they are self-irrigating and self-fertilizing with established vegetation. Also, they provide human comfort by cooling the local microclimate through shading and evapotranspiration.

15 Terminology for stormwater infrastructure varies considerably across different reference materials. In the *Portland Stormwater Management Manual,* this type of facility is simply termed a "basin."

16 Woods-Ballard, et al., *The SuDS Manual*, 361.

17 Woods-Ballard, et al., *The SuDS Manual*, 33.



//Figure 2.6

Typical section of a bioretention facility. Also displays processes of runoff reduction. (GeoSynctec Consultants. https://www.geosyntec.com/consultants/publications/69-publications/6224-sediment-bioremediation).

//Chapter 3 Preview

The chapter ahead follows the three-phase design portion of this project. It has been noted before—but is worth reiterating—that the primary "design" in this project's approach is not the direct arrangement of landscape elements based on subjective decisions. Instead, the design is of the generating system—the algorithmic model—that automatically produces site-wide stormwater management designs. As outlined in Section 2.2.iv, the phases divide the full-extent of the algorithmic development along the synthesized framework sequence. Phase 1 aligns with the representation stage, Phase 2 follows the search process for design solutions and Phase 3 is an evaluation of the algorithmic model and its solutions.

"The new primitives are animate, dynamic, and interactive entities—splines, nurbs, and subdivs—that act as building blocks for dynamic systems..."

—Patrik Schumacher[1]

1 Schumacher, Patrik."Parametricism as Style—Parametricist Manifesto." Lecture at 11th Architectural Biennale. Venice, Italy, 2008.

3 Algorithmic Design

3.1//Phase 1: Representation

This phase of the algorithm development covered an extensive series of steps. It involved modeling the site in Rhino+Grasshopper, simulating hydrological conditions, and setting up the various constraints for the solution domain used in the Phase 2 search process. Fundamentally, this was a process of formulating a problem in which the spatial properties of the site form the variables. The solution(s) to this problem—the optimized bioretention facility model (BFM)—is also spatial in nature, and therefore defining the solution domain is a matter of delimiting a space in which the BFM can be systematically tested.

Put broadly, there were two types of procedures that occurred in this phase—those that occur in 3D and those that are planar (Figure 3.1). Finding the solution domain in 3D would be needlessly complex and error-prone, especially when making topological changes to solid models such as the terrain mesh. Whereas finding the solution domain in planar space both eliminates the extra possibility of error and drastically increases processing time.[2]

The following sections are simplified synopses of the algorithm coding steps, including retroactive assessments of the steps taken.

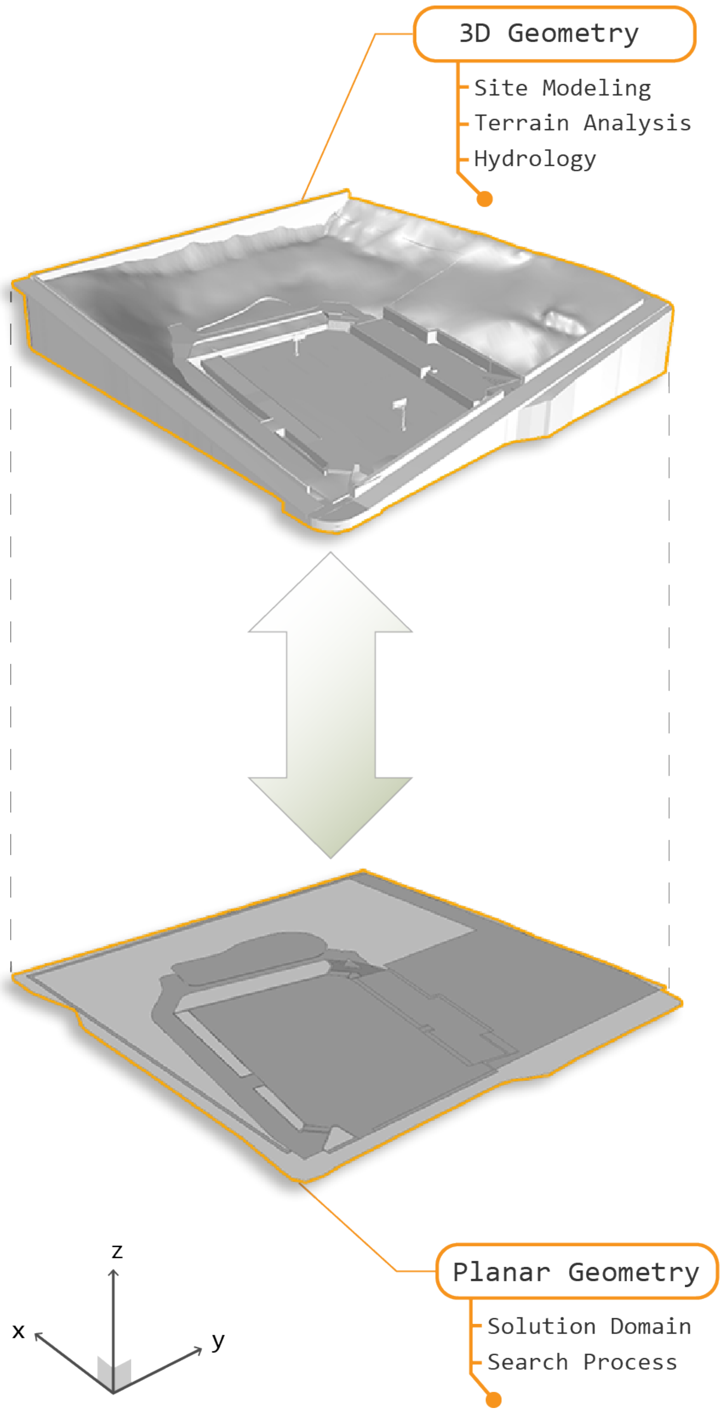
2 Initially, a solid model with vertical dimensions of the site was required for analyses of terrain and hydrology. Once calculated, that data was easily projected onto a planar surface—the xy-plane in model space—on which the solution domain for the BFM was ultimately defined.

i. Setup Site Model

Multiple data sources were used in attempts to construct an accurate mesh of the site terrain—not all were met with success. The first attempt involved employing a quadcopter drone with gimbal-mounted, wide-angle lens camera to collect imagery of the site.[3] The resultant model included a remarkably high-resolution mesh as well as an overlaid 3D image (Figure 3.2). However, this mesh was ultimately unusable because of the many imperfections in the terrain depiction.

Progress in creating the base mesh was reached in using CAD drawings[4] shared by GreenWorks. The topographic 2-ft contour lines from the drawing provided a frame on which to build a non-uniform basis spline (NURBS) surface[5]. Converting the NURBS surface to a mesh was a straightforward process in Rhino which yielded a mesh with a polygon count of approximately 10,000 (Figure 3.3). Compared with the drone mesh, the chosen mesh has a lower resolution by a factor of 10. However, the lower resolution proved necessary for adequate digital processing time of spatial analysis steps.

From the mesh it was possible to



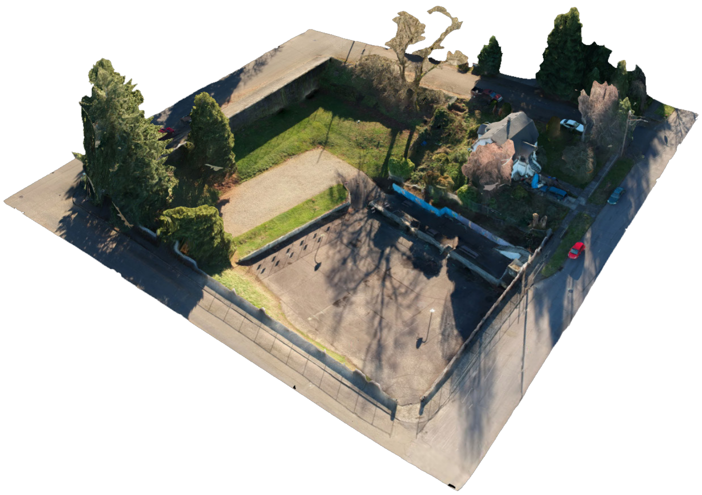
//Figure 3.1

The two types of procedures in this phase use either planar or 3D geometry.

3 A series of 157 photos were taken along a preprogrammed flightpath at an altitude of 200 feet above the site. These photos were submitted to the service DroneDeploy which uses proprietary image processing software to produce a photogrammetric model of the site.

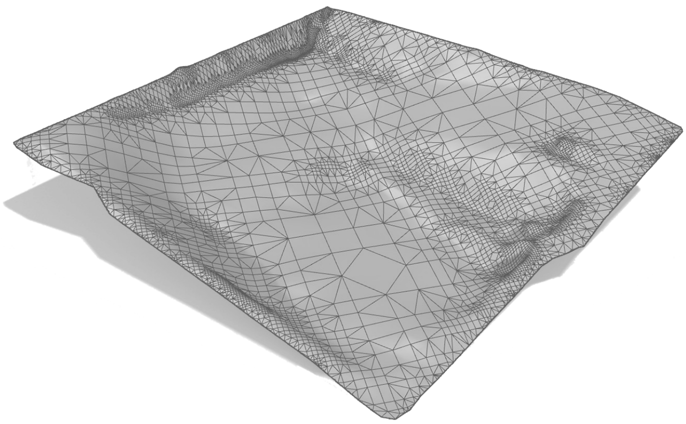
4 Halpin, Margo and Johnson, Ben. “Latourette Park 30% Design,” 2019.

5 Non-uniform rational basis spline (NURBS) is a type of 3D modeling geometry that has a higher degree of control and resolution for curves than mesh modeling.



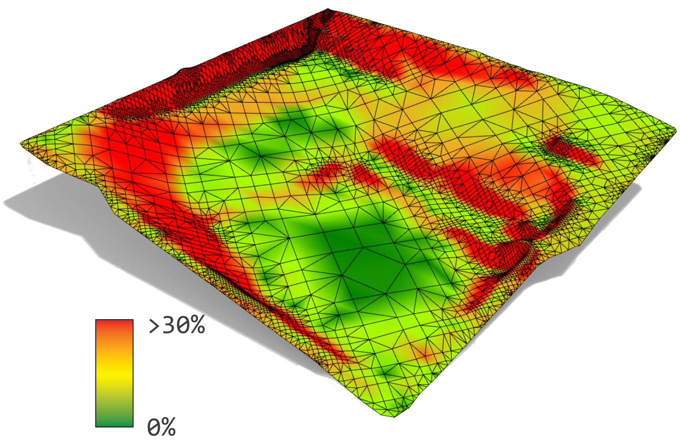
//Figure 3.2

Photogrammetric model from drone photography.



//Figure 3.3

Final base mesh representing site terrain.



//Figure 3.4

Slope map of the base terrain. Red is classified as excessive slope.

determine slope conditions throughout the site. In Rhino this is achieved by measuring the angle of each face of the mesh relative to the xy-plane of the model space. The results of this process may be viewed as a heat map in Figure 3.4.

Once the mesh was resolved the next step was to built objects on its surface. An important assumption made at this stage was that certain elements in the design were to remain fixed out of practical necessity. I learned through conversations with experts that much of the leftover structure of the former pool would remain. The concrete platform and retaining wall behind it were considered passable in their current condition and much of the asphalt area was to remain paved, although resurfaced.[6]

The new design would attempt to significantly reduce paved area, but an area of approximately 10,000 ft^2 would be fixed. Additionally, the retaining wall which separated the lower paved area from the terrace level was to remain largely unchanged.

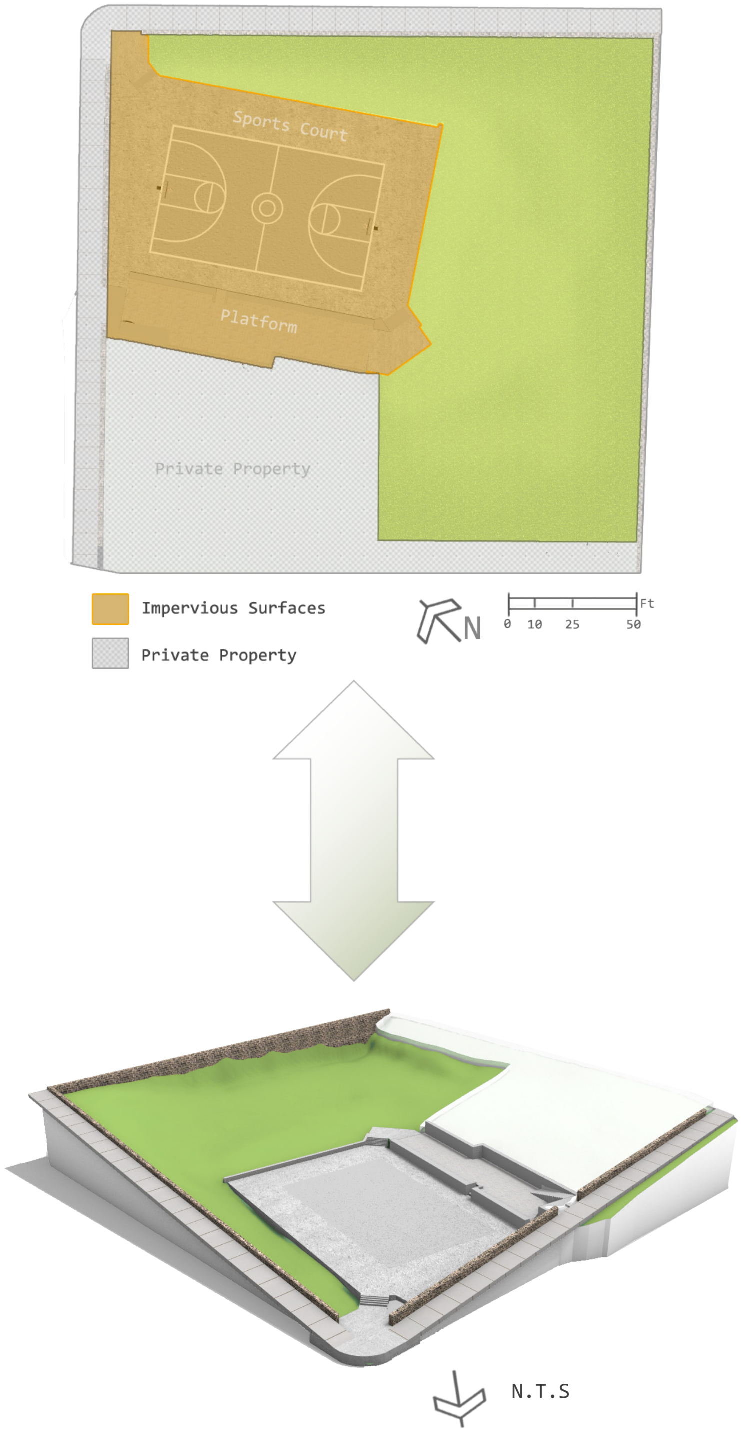
6 It is unknown whether the former pool basin was demolished and removed before the site was filled in 1969. Therefore, the design would progress with the cautious presumption that the pool was buried, and the surface above it would be kept impervious.

These existing objects were added to the model through a transposition from the CAD drawings to the surface (Figure 3.5). The definitive limit-of-work was also obtained from GreenWorks—thus taking into account the existing conditions in the southwest corner that were to remain.

ii. Hydrological Analysis

The first step of the hydrological analysis determined the direction of surface flows and identified the end-points of those flow paths. Essentially the goal was to create a hydrological map displaying how water moves across the site and where the drainage occurs.

During the research of Rhino +Grasshopper, it was discovered that other users have pioneered techniques for site-level hydrological mapping. Of the options available, the plugin Groundhog[7] was found to feature the most capabilities and produce the best data. Its tools for drawing flow paths present the user with controls to finesse the granularity of results, letting the user scale the process according to site size and expectations of accuracy. The tool works by first simulating rainfall on the site by randomizing the placement of hundreds of points on the mesh (Figure 3.6.a). Working down from those randomized starting points, flow paths are drawn in individual segments (Figure 3.6.b) until a low-point is reached[8] (Figure 3.6.c).



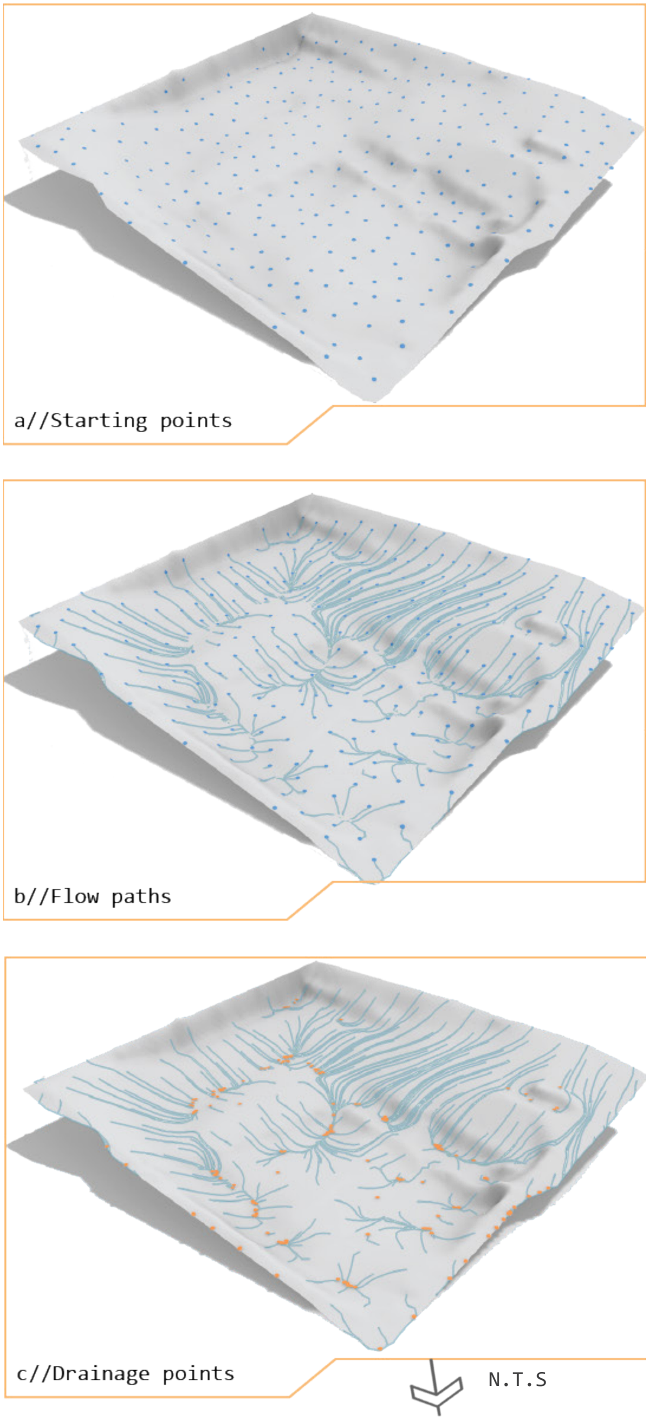
//Figure 3.5

Fixed boundaries for alterable surface area. Limit-of-Work and impervious surfaces are off-limits area.

---

7 Groundhog was developed and copyrighted by Philip Belesky. (Source: https://www.groundhog.la/).

8 From each flow segment's starting point on a particular mesh cell, the neighboring cells are evaluated to determine which one poses the path of least resistance to surface flow.



//Figure 3.6

Simulated flow paths over site terrain. Generated with the Groundhog plugin.

Upon determining the flow paths, the next step was to divide the site into sub-catchments. Doing so allows for an extra degree of localization of results. This turned out to be a crucial step with the Latourette Park site, as there was significant variation in slope and land cover conditions between the resultant subcatchments. Factoring those conditions separately provided the opportunity to generate custom BFM solutions for each subcatchment in later steps. Groundhog was used again for this process of delineation (Figure 3.7). Results were the creation of three prominent sub-catchments whose division appeared to be caused by flows converging on several existing footpaths and stairwells.

With the spatial information in order, a numeric simulation of the target design storm was made. To reiterate the parameters of the design storm, the PSMM specifies that stormwater interventions should be sized to handle the runoff from a 10-year storm. It further specifies that the drawdown time[9] (DDT) should not exceed 30 hours—a parameter which comes into play in later steps.

The point of this stage of the hydrological analysis was to determine exactly how much runoff needed to be treated in each sub-catchment. Runoff quantity is largely dependent on two factors: the intensity of rainfall and the physical characteristics of the catchment. As prescribed by the *SuDS Manual*, the Modified Rational Method (MRM) is used to relate these factors together. The MRM equation is as follows:

9 *Drawdown time* refers to the timeframe necessary for a stormwater facility to completely withdrawal its runoff content after a designated storm event.

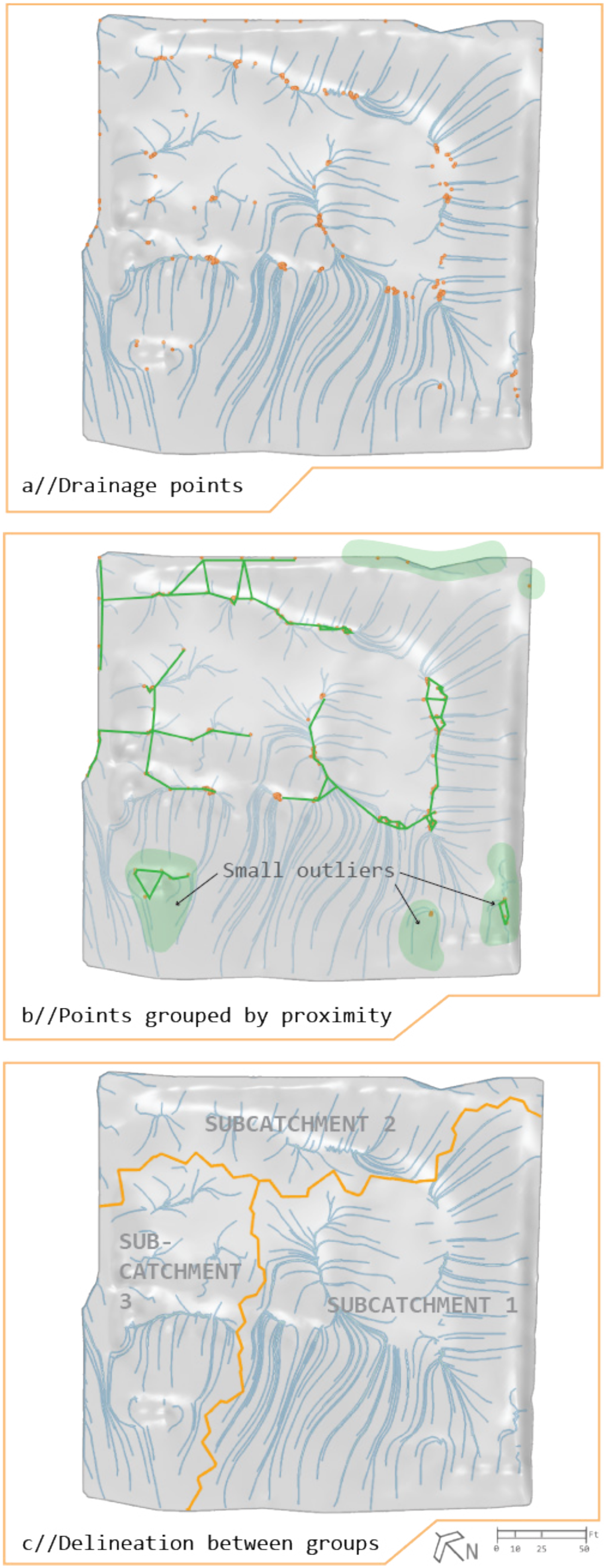
$$Q = C*C_A*i*A$$

where:

Q = Peak runoff rate (ft^3/sec)
C = Coefficient fo runoff (see below)
C_A = Antecedent factor[10]
i = Rainfall intensity (in/hr)[11]
A = Catchment area (acres)

Coefficient of runoff is a mathematical factor that approximates what proportion of rainfall remains as runoff after having infiltrated the ground. It is generally determined by three site conditions: hydrologic soil group, average slope, and land cover. The USDA Natural Resource Conservation Service (NRCS) classifies the entire site as having Woodburn silt loam—which falls in group B of hydrologic soils. Land cover and average slope varies across sub-catchments.

In finding the runoff value, another factor is the time of concentration (T_C). This value is the time that it takes for the longest course of surface flow to reach the lowest point in the catchment. There are a variety of methods in determining this

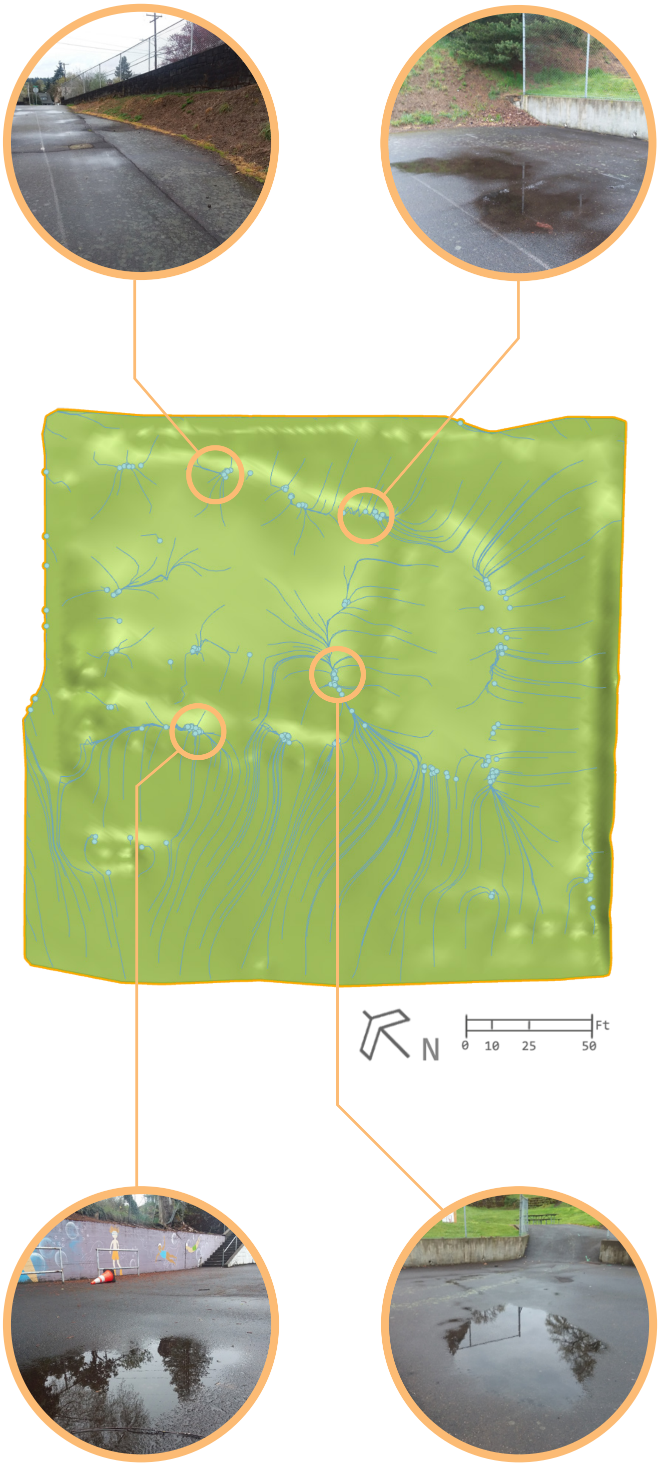


//Figure 3.7

Subcatchment delineation. (a) Starting from the previously determined flow paths and drainage points, (b) this process works by grouping neighboring drainage points together according to a user-set proximity threshold between drainage points. (c) A boundary is then drawn between the respective flow paths of the separate groups. Small outliers are subsumed within larger catchment areas.

10 The antecedent precipitation factor (C_A) is based on the assumption that sustained storm events produce more runoff over time as the pervious ground becomes saturated and infiltrates less.

11 Rainfall intensity is determined graphically from a rainfall intensity chart and corresponds to the storm frequency. See Appendix A, Table A1 for rainfall intensity data.



//Figure 3.8

Ground truthing of drainage conditions onsite. Water pooling on the asphalt corresponds with dense clusters of flow paths from the hydrological analysis.

and there is no specific method endorsed by the SuDS Manual. The equation known as the Federal Aviation Administration (FAA) Method was selected to calculate time of concentration. That formula is:

$$T_c = G(1.1 - C)L^{0.5} / (100 * S)^{1/3}$$

where:

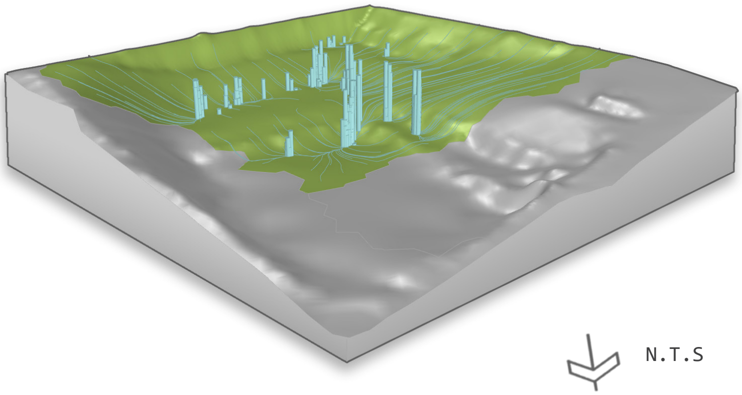
T_c = Time of concentration (min)
G = Constant of 1.8
L = Longest flow path (ft)
S = Average slope in catchment

Using the results of both the Q and T_c, it was then possible to determine the volume of runoff using a hydrograph, which is a plot of flow rate (Q) versus time.

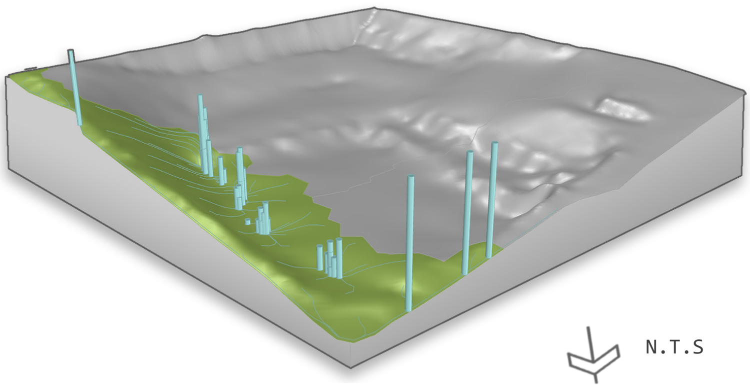
Each of the subcatchments was analyzed with these methods, ultimately producing catchment-specific values of runoff volume and flow direction. To add an extra degree of localization, the runoff volume reaching each low-point was calculated.[12] Within each subcatchment there were areas where runoff collection was relatively concentrated. Visualization of this condition showed where runoff was consolidating and subsequently where stormwater interventions could be made for the highest localized effect. Quantifying and visualizing this quality proved extraordinarily useful in siting and sizing BFMs. It permitted the siting of multiple stormwater facilities within the subcatchments—each with their own unique targets for runoff reduction.

Following outcome of the hydrological

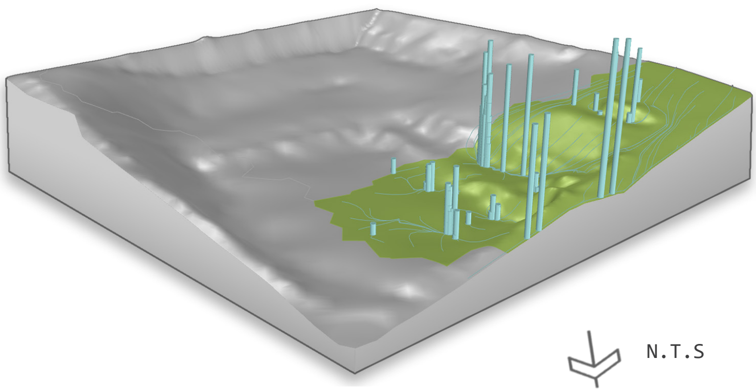
12 This was accomplished by taking the proportional length of each flow path and multiplying it by the total runoff volume of the subcatchment.



Subcatchment 1	
Area	28,134 ft^2
% Impervious	18%
Average Slope	23°
Coefficient of Runoff	0.42
Runoff Volume (ft^3)	3,348 ft^3



Subcatchment 2	
Area	10,287 ft^2
% Impervious	0%
Average Slope	13°
Coefficient of Runoff	0.42
Runoff Volume	1,224 ft^3



Subcatchment 3	
Area	13,549 ft^2
% Impervious	36%
Average Slope	17°
Coefficient of Runoff	0.65
Runoff Volume	2,386 ft^3

//Figure 3.9

Hydrological properties of the three subcatchments within Latourette Park. The columns represent the proportional volume of runoff reaching flow path drainage points. Height and density of column clusters generally correlate with the need for targeted intervention along the associated flow paths.

analysis, a site visit was made after a rain event to confirm drainage conditions. It was observed that terrain and hydrological conditions generally matched the simulated conditions, as evidenced by the pooling of surface runoff at the expected collection points (Figure 3.8).

Figure 3.9 presents the hydrological properties of each subcatchment, including a visual representation of relative flow volumes as depicted by the columns in each image.

iii. Setting Up the Solution

The previous step identified where surface flow was going and in what quantities. Based on that information, the next step was to determine where stormwater interventions were to be made for greatest efficacy.

In terms of developing the algorithm, this step created the essential parameters of the *solution domain*[13]—which is simply the boundaried space that potential solutions were generated within. They defined where BFMs were to be centered and what spatial extent was permitted for the BFM surface shapes. The algorithm would use these constraints in the Phase 2 search process as the range of possibilities for BFM solution configurations—automatically nullifying search results that exceed the domain space. Defining the solution domains was done through the steps discussed ahead, which are summarized in Figure 3.10.

13 *Solution domain* refers to the range of possible solutions that are produced in the search process. In this modeling scenario it is represented spatially.

Initial Placement of Solution Domains

The first step in finding this domain was mainly accomplished with the Grasshopper metaball tool[14]. This tool automatically created an amalgam of shapes that roughly correspond in area to the surface area needed for BFMs. These amalgamated shapes were centered over the drainage points (Figure 3.10, part A). A multiplier was then applied to approximate the solution domain within a spatial extent grounded in stormwater management guidelines for sizing facilities.[15]

The PSMM offers a simplified sizing factor of 0.09 to be used in approximating infiltration basins' areal coverage.[16] Based on this figure, it was reasoned that a solution domain equivalent to 12-15% of the subcatchment area would satisfy the broad guideline of 9% while giving a buffer space for various surface shape configurations of the BFM to be attempted.

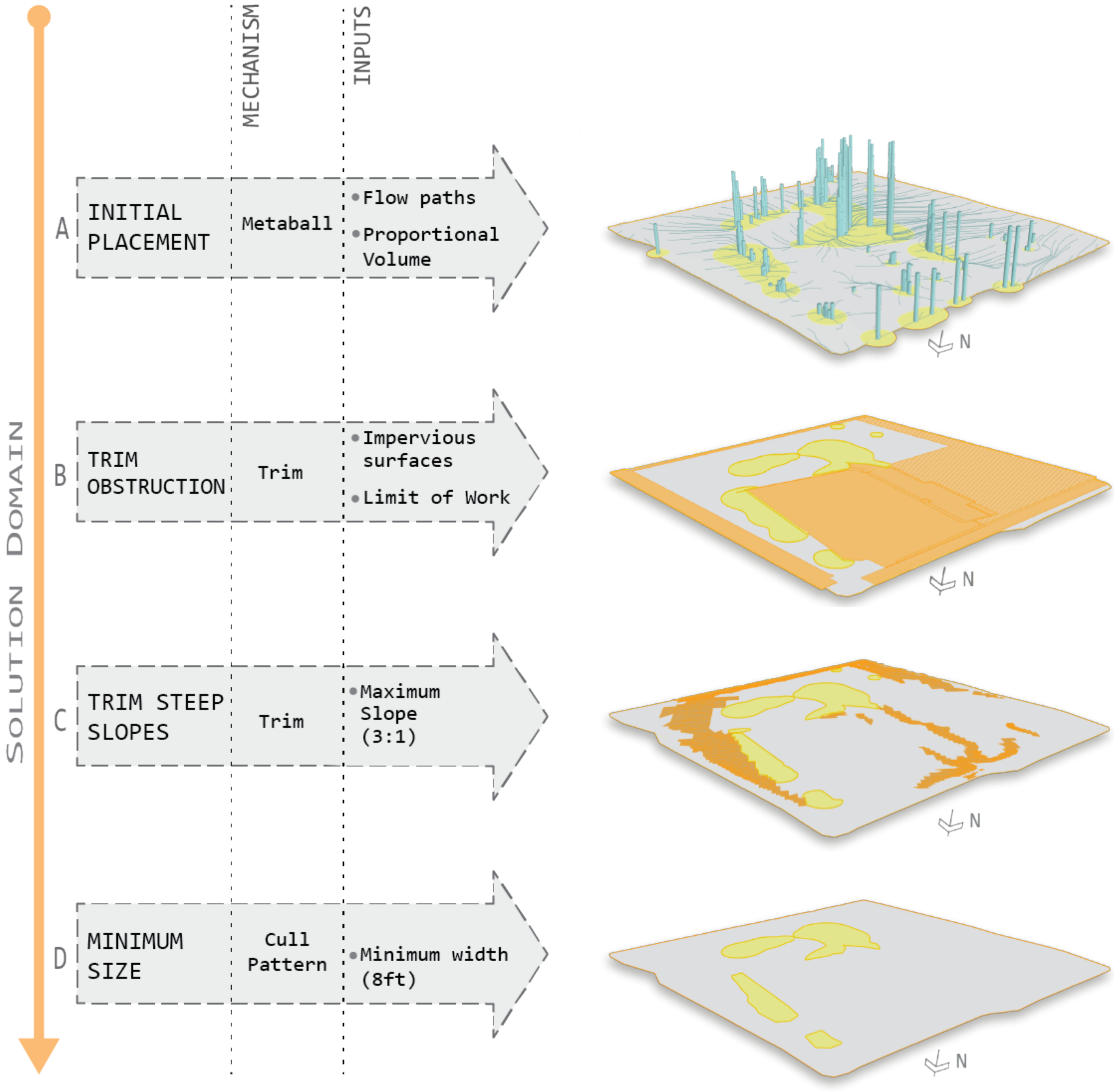
Trim Solution Domains

Following the initial placement, the solution domain was limited to only available spaces. It had to exclude surface areas

14 The metaball tool works by generating variously sized circles around a series of designated center-points, and then blends those circles into a single shape. This algorithm used flow path end-points as center-points and the circle radii directly related to the proportion of flow reaching that point.

15 The multiplier was applied to the radii of the individual circles that form the metaball combined shape.

16 This is a broad approximation for sizing small-site stormwater management designs. It stipulates that at least 9% of the catchment area be designated for stormwater infrastructure.



//Figure 3.10

Algorithmic process sequence of placing, sizing and constraining the solution domain.

that were off-limits due to project boundaries or surface materials (Figure 3.10, part B) and areas with excessive slopes (Figure 3.10, part C). The threshold value for each of these criteria was defined with adjustable parameters and then calculated through the algorithm. Excessive slopes were identified as any one exceeding a 3:1 ratio—the specified maximum allowance for sidewall slopes per PSMM bioretention facility BMPs.[17] Other excluded areas were the fixed impervious surfaces[18] and the area past the limit-of-work.[19]

Minimum Size Requirements

Another general measure of feasibility was the minimum size required for a facility to be worth constructing. Smaller solution domains were removed by a culling pattern that selected and eliminated ones under the threshold set by the adjustable parameter (Figure 3.10, part D). With the Latourette Park model, only exceptionally small solution domains were removed. These were the spaces that would have been too small or narrow to construct a basin with a level bottom at least two feet in width.

Edge Smoothing

Some of the boundary edges were made jagged by the trimming process, whereas a smooth edge would be ideal for the solution search process. To correct these edges, a simple python script[20] was executed to remove shorter line segments and redraw the shape of the solution domain. The final result was a set of simplified solution domains that corresponded to where BFMs could be built for maximum effect.

17 Portland BES, "Portland Stormwater Management Manual," 2-86.

18 Eliminating excessive slope required a culling technique that selected mesh cells designated by the threshold value. Selected mesh cells were then projected to planar space and used as a trimming object against the solution domain.

19 A similar procedure was employed to trim away the limit-of-work. The model layers representing these surfaces were selected, projected to planar space, and used as trimming objects.

iv. Bioretention Facility Model Definition

Having finalized the solution domains, the next step was to create the representation of the BFMs. These were to be their own 3D objects so that volumetric performance calculations could be made. Their completed form needed to represent the dimensions of the facilities as they would occur on the ground, in compliance with local design BMPs. Therefore, the PSSM was referred to for such key dimension values as: freeboard depth, sidewall slopes, filter medium depth, and drainage layer depth (Table 3.1).

Each of these specifications were provided within a range of acceptable values; however, the maximum value for each was chosen. This crucial choice was based on the assumption that doing so would significantly simplify the Phase 2 search process. The aim was to maximize volumetric performance and significantly shorten the runtime of the solver by reducing the range

20 Grasshopper allows the integration of python scripts as custom components.

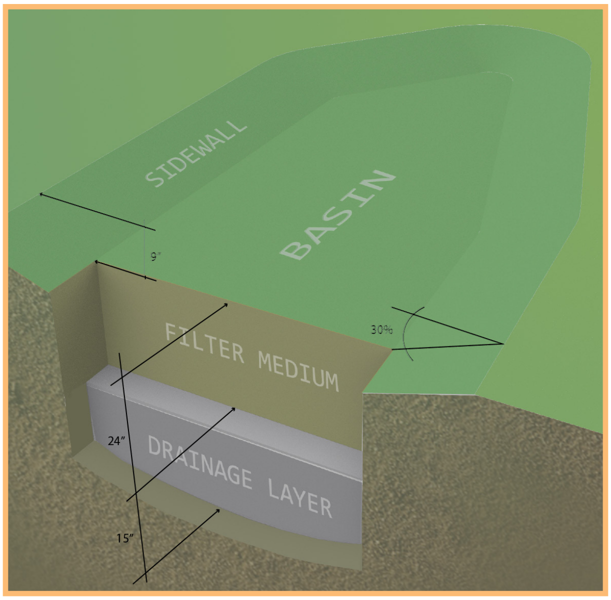
of possibilities for the solution domain.[21]

Create Surface Shape

Modelling the BMP dimensions was relatively straightforward in Grasshopper. The surface shape was created by first offsetting to the inside of the solution domain and dividing the result into multiple curve segments[22]. Corners were set to automatically filet to a 3° radius.[23]

Basin Formation

Next the basin of the BFM was formed. For simplicity, the basin was approximated as a trapezoidal depression with a level bottom. Using the BMPs for freeboard depth and sidewall slopes (Table 3.1), the linear length of the sidewall was calculated trigonometrically. The next step was to form the filter and drainage layers below the floor of the basin. This was accomplished by simply extruding the floor shape for the predetermined depths of those layers, which was 24 inches and 15 inches, respectively (Table 3.1). Each of these layers are tied together so that any changes made to the surface shape of the basin would automatically alter the formation of



//Figure 3.11

Cross-section of generic BFM form as determined by local BMPs, as listed in Table 3.1

PSMM Bioretention Design Guidelines	
Freeboard Depth	3" - 9"
Sidewall Slope	Max of 3:1
Filter Medium Depth	18" - 24"
Drainage Layer Depth	12" minimum +3" transition layer

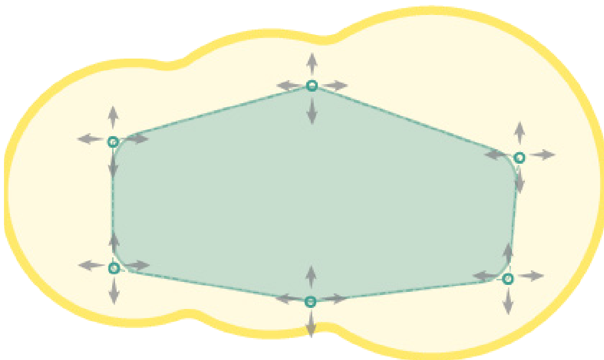
//Table 3.1

BMPs for bioretention facility dimensioning. (Data adapted from Portland BES, *Portland Stormwater Management Manual*, 2-87.)

21 An extra variable for depth would needlessly prolong the search process as the optimal value for the depth factor would be fairly deterministic. The search process would ultimately find that a greater depth equals more runoff reduction without impacting other performance metrics—and would therefore max out the depth variable.

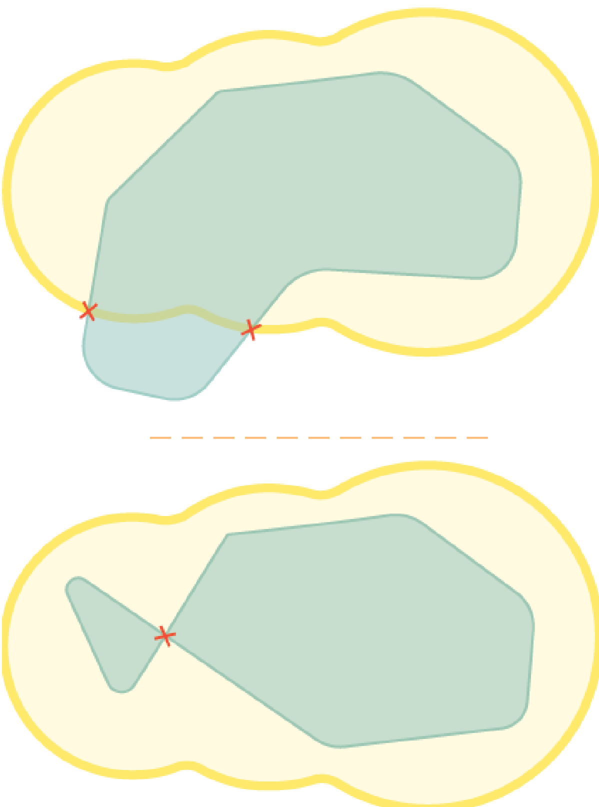
22 Segments were created at a set length of 20-ft. The number of segments was therefore determined by the length of the original curve.

23 Corners were fileted to approximate a realistic construction shape. Otherwise acute angles would have formed in the corners.



//Figure 3.12

Different solution generations were accomplished by giving the vertices of the BFM a range of movement within the domain.



//Figure 3.13

(Top) Example of a null solution caused by the BFM exceeding the solution domain and (Below) a null solution caused from a self-intersecting BFM.

the entire BFM. At this point, the algorithm was provided a default BFM (Figure 3.11) that was roughly congruent to the solution domain.

Variable Surface Shape

The difficult aspect of this step was in enabling the BFM to be reconfigured into a variety of shapes. The default shape described in the preceding steps is merely the template for potential solutions, and it is through a series of reconfigurations that the search process finds an optimized solution.

Several approaches were attempted before finding a viable method of altering the BFM shape. That method worked by giving the vertices of the surface shape their own independent range of 2-dimensional movement (Figure 3.12). Adjustable parameters were added for both the x and y direction of movement. The Phase 2 search process would then produce different outcomes by generating different values for those vertex movements.

Throughout the trial and error of this method, setbacks were encountered with inadvertently generated shapes that were wholly impractical. BFM shapes had occasionally twisted themselves into multi-lobed shapes, and/or exceeded the spatial extent of the solution domain. To correct this dilemma, conditional statements were included to automatically nullify solution outcomes where BFM boundaries intersected either the solution domain or itself (Figure 3.13).

3.2//Phase 2: Search Process

At this point in the algorithm development, the solution domain and BFM were completely defined as parameters for the search process. However, the solution search process also requires a precise definition on how to measure the performance of each BFM solution. Specifically, those performance metrics are the measures of *cost*, *runoff reduction* and *targeted interception* of surface flows near their source. The sections ahead describe how these conditions are both measured and accounted for.

The search process itself is performed with the Grasshopper plugin Galapagos[24]. It is an evolutionary-solver [25] that uses principles of natural selection to narrow in on the best-fit solutions. Each of the search process steps of generation, evaluation, and guidance are essentially built into Galapagos. Briefly put, Galapagos performs a series of solution iterations that improve progressively using each solution's fitness value[26] as a basis for its quality. The fitness value is explicitly defined through an expression written by the designer and executed through the algorithmic model. The overall purpose of Phase 2 was to arrive at that fitness expression by defining and relating the performance metrics that are the expression's variables. For an explanation of the mechanics of this process, please refer to Appendix B.1.

i. Measuring Runoff Reduction

Hydrologic engineering is an immensely complicated practice with an evolving assortment of techniques for understanding how water moves and how to design for stormwater solutions. The actual sizing of facilities is a complicated matter with a multitude of options for formulating solutions. In this design scenario, the overarching objective was to reduce runoff by means of infiltration. Other methods of runoff reduction—such as on-site storage and offsite drainage—were not options to consider. The best technique discovered was sourced from the SuDS Manual.[27] It determines the volume of infiltration that occurs through the sides and bottom of the bioretention facility into the surrounding soil. The volume infiltrated through the bottom of the basin is given by:

$$V_{infB} = A_B * DDT * I_R / 12$$

where:

V_{infB} = Reduction at bottom (ft^3)
A_B = Surface area at bottom (ft^2)
DDT = Drawdown time (hr)
I_R = Infiltration rate (in/hr)

24 This plugin was developed by Grasshopper creator David Rutten and was released in 2007. Since its inception, it has been applied to a wide variety of computational design problems.

25 *Evolutionary-solvers* are algorithms that are used to find near-optimal solutions when it is not possible to find the optimal solution through deterministic, linear functions. They work off the biological principle of natural selection, starting with a large population of potential solutions and gradually culling out poor solutions through many progressions.

26 *Fitness Value* is the Galapagos term for the overall quality of a given solution. It is determined by a mathematical expression that relates each of the performance metric variables.

27 Woods-Ballard et al., *SuDS Manual*, 342.

While the volume of infiltration through the sides is expressed as:

$$V_{infS} = (A_O - A_u)*DDT*I_R/12$$

where:

A_O = Area of basin top (ft^2)
A_u = Area at underdrain (ft^2)

Total volume of runoff reduction is attained by adding these figures together:

$$V = V_{infB} + V_{infS}$$

In this context, the infiltration rate (I_R) is that of the surrounding soil. That figure correlates to the soil's hydrological group, which was Group B at the Latourette Park site. Determining the precise value of the infiltration rate is ideally a matter of field-testing soil properties on-site. But for lack of that precision assessment, a table of values was referred to for the infiltration rate (Appendix A, Table A2).

These expressions were written into the algorithm and tied to the appropriate dimensions of each of the generated BFMs. The resulting value for runoff reduction was later factored into the fitness expression as RR_{bfm}.

ii. Measuring Cost

Following established models[28], cost was measured as the construction costs plus the cumulative cost of maintenance over the lifetime of the facility. The lifetime was set at 30 years based on the typical service period of stormwater facilities.[29] Both construction and maintenance were calculated using cost estimates obtained from experts. GreenWorks provided the figure for average construction costs, which was \$50 per ft^2 for concrete-lined facilities \$15 per ft^2 of bioretention facilities.[30] This figure factors the cost of materials and related rates, including: excavation/ disposal, importing soil, erosion control, plants, irrigation, pipe, overflow drain, stormwater network connection fees, and concrete. It does not factor the cost of labor. Maintenance costs were retrieved directly from the Oregon City Pubic Work.[31] They reported that the current rate billed to maintenance contractors is \$0.25 per ft^2 annually. Within the algorithm, total cost was calculated as:

$$C = A_{BFM} * (C_C + 30*C_M)$$

where:

A_{BFM} = Surface area of BFM (ft^2)
C_C = Cost of construction (\$)
C_M = Maintenance cost rate (\$)

iii. Measuring Targeted Intervention

The algorithmic search process was designed in order to favor BFM solutions that received the most runoff input as a result of their spatial configuration. Determining this condition was a matter of detecting where the BFM solution boundaries intersected or encompassed flow paths and/or

28 Chui, Ting et al. "Assessing Cost-Effectiveness of Specific LID Practice Designs in Response to Large Storm Events." *Journal of Hydrology*. Vol 533 (2016): 357.

29 Ibid, 353.

30 Ben Johnson, email to author, April 11th, 2019.

31 Eric Hand, Oregon City Dept. of Public Works, email to author, April 13th, 2019.

their end-points.[32]

Subsequently, the volume of runoff was calculated for those inflows. In the event of an end-point falling within a BFM, the entire runoff volume of its respective flow path was counted. The inflow for flow paths that only intersected the BFM was calculated in proportion to length of flow path from above the BFM. The total amount of inflowing runoff volume was then used as a fitness parameter, thereby setting a custom target for runoff reduction for each of the BFMs.

iv. Fitness Expression

The purpose of the fitness expression was to combine the values for each of the preceding fitness objectives into a single value to be evaluated by the evolutionary-solver, Galapagos. Based on a generated solution's fitness evaluation, Galapagos could then adjust the BFM shape in the next iteration in order to progressively improve fitness values. That was the essential mechanism of the search process—a cyclical process of generating solutions, evaluating their fitness, and adjusting the subsequent solution.

It becomes a challenge, however, to define a single, all-encompassing fitness value when the variables essentially oppose each other. Such was the case between the fitness objectives of cost and runoff reduction—as one improved, the other worsened. A similar dynamic complicated the fitness objective for targeted intervention. The value for this objective increased as more runoff inputs were added to the BFM. But this also resulted in a changing target for runoff reduction and consequently a higher cost. When counted together, each of the fitness objects pull the solution in different directions.

Different expressions were written to experiment with solution outcomes and, ultimately, the winning strategy was to relate the objective variables to each other in order to produce the lowest possible number. This may seem counter-intuitive, but Galapagos has the option to optimize for either the highest, or lowest fitness value. In this algorithm, Galapagos was duly setup to optimize solutions to the lowest fitness value. The basic expression for this strategy was:

Runoff Reduction + Cost - #Inputs

In setting up the expression this way, the best fitness score results from solutions with a runoff reduction that most closely matches the reduction target, the lowest cost, and having the highest number of runoff inputs. A more precise definition of the expression reads as:

$$F = abs(RR_{tar} - RR_{bfm}) + C - (I_{pt} + I_{fp})$$

where:

F = Fitness value

abs = absolute value (notation)

RR_{tar} = Reduction target (ft^3)

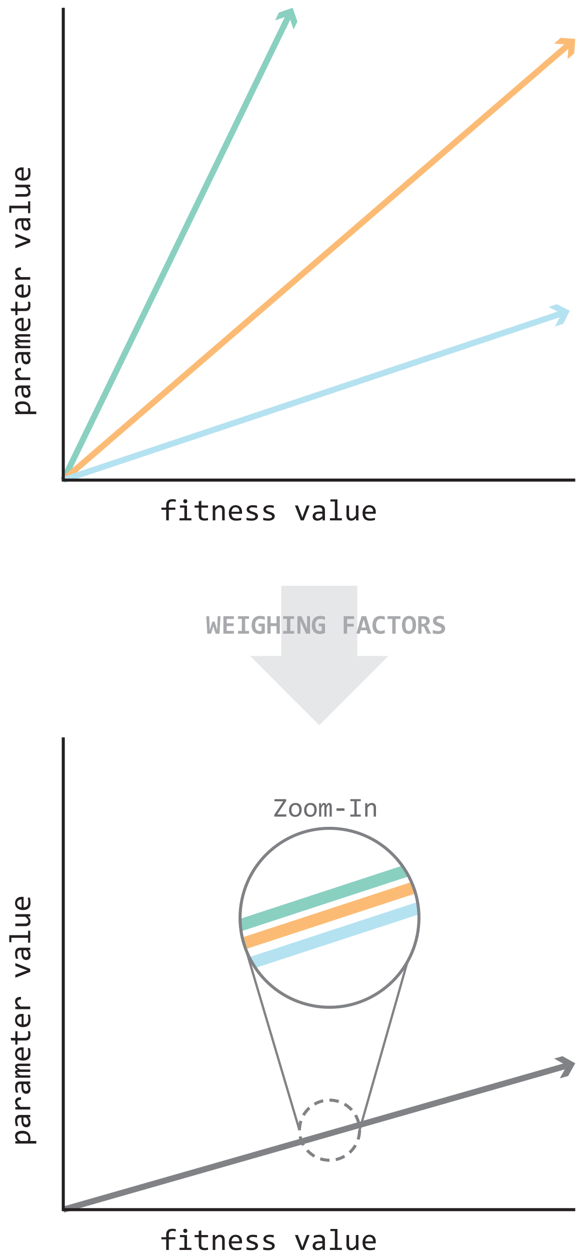
RR_{bfm} = Reduction obtained (ft^3)

C = Total cost ($)

I_{pt} = Number of drainage points in BFM

I_{fp} = Number of flow paths passing through BFM

32 This operation was performed by projecting the hydrological data onto planar space where it was possible to automatically highlight instances where that geometry intersected the surface shape of the BFM. For each BFM, the number of inflow inputs was counted—with care not to double count flow paths with end-points within the BFM.



//Figure 3.14

Generic example of normalization. Weighing the values of fitness objective variables weighs balances their influence on the total fitness value.

The previous expression has the essential logic defined, however a problem results from the objective variables being factored equally. That is a problem because each of the objectives has different ranges in value—meaning they have vastly different influences on the solution. For instance, cost may be expressed somewhere between 1,000 and 15,000 while the difference of runoff reduction could be reduced to a single-digit value. In order for these values to give a balanced outcome they had to be weighted to put all of the objectives' values within a similar range—they had to be *normalized* (Figure 3.14). In this way, each objective variable is brought close to one.[33] The final, normalized version of the fitness expression was written as:

$$F = abs(RR_{tar} - RR_{bfm}) + \frac{C}{C_{MAX}} - \frac{(I_{pt} + I_{fp})}{I_{MAX}}$$

where:

abs = absolute value (notation)

F = Fitness value

C_{MAX} = Range in cost

I_{MAX} = Range in # inputs

33 *Normalization* is the process of adjusting value ranges of different data sets to a common scale. Values were normalized by dividing the variable by the maximum range of its value. That range was determined by testing the largest possible BFM within each of the solution domains.

3.3//Phase 3: Solution Assessment

The following section examines the efficacy of the algorithmically generated solutions in an effort to assess the feasibility of this approach within the overall site-design process. It is compared against the design put forth by GreenWorks—both of which are evaluated in terms of runoff reduction and cost-effectiveness. Based upon this evaluation, recommendations are made on how to proceed with the algorithmic approach in Section 3.4.

i. Algorithmic Solutions

Throughout the previous two phases of the algorithmic approach, the site was subdivided into three distinct subcatchments, and subdivided again into separate BFM solution domains. In this way, the stormwater intervention proposed within each solution is custom fit to its immediate context within the site. As displayed in Figure 3.15, four different BFM solutions were ultimately generated in this process. Two were generated in Subcatchment 1, two in Subcatchment 2, while none were generated in Subcatchment 3. Each was in response to its own unique configuration of localized constraints and specific runoff reduction targets.

The following breakdown addresses the BFM solutions individually, by order of the subcatchment they occur within. Included with each are the definitions of the solution domain and the final BFM configurations.

//Figure 3.15

Algorithmically generated results for solution domains and BFM solutions separated by subcatchments.

Subcatchment 1: BFM-1A

Solution domain: This first BFM solution accomplished the majority of the site-wide runoff reduction—successfully servicing approximately a quarter of the entire area. It was placed at a critical convergence point, where flows from the steeply sloped sides on the south edge meet on the terrace level. Given the high amount of inflow, the algorithmically determined solution domain was considerably large. However, that domain was also highly constrained by landcover conditions. The initial solution domain of 5,282 ft^2 overlapped the transition between the top terrace level and the paved terrace below. After trimming away the part of the domain that fell within the impervious landcover, the solution domain reduced to 2,433 ft^2. This trimming step also managed to create an awkwardly shaped solution domain that featured two large protuberances off the central mass of the shape. Consequently, the solution search process for this BFM was significantly hampered—as the articulated edge resulted in more instances where segments of the BFM solution shape intersected the boundary of the solution domain.

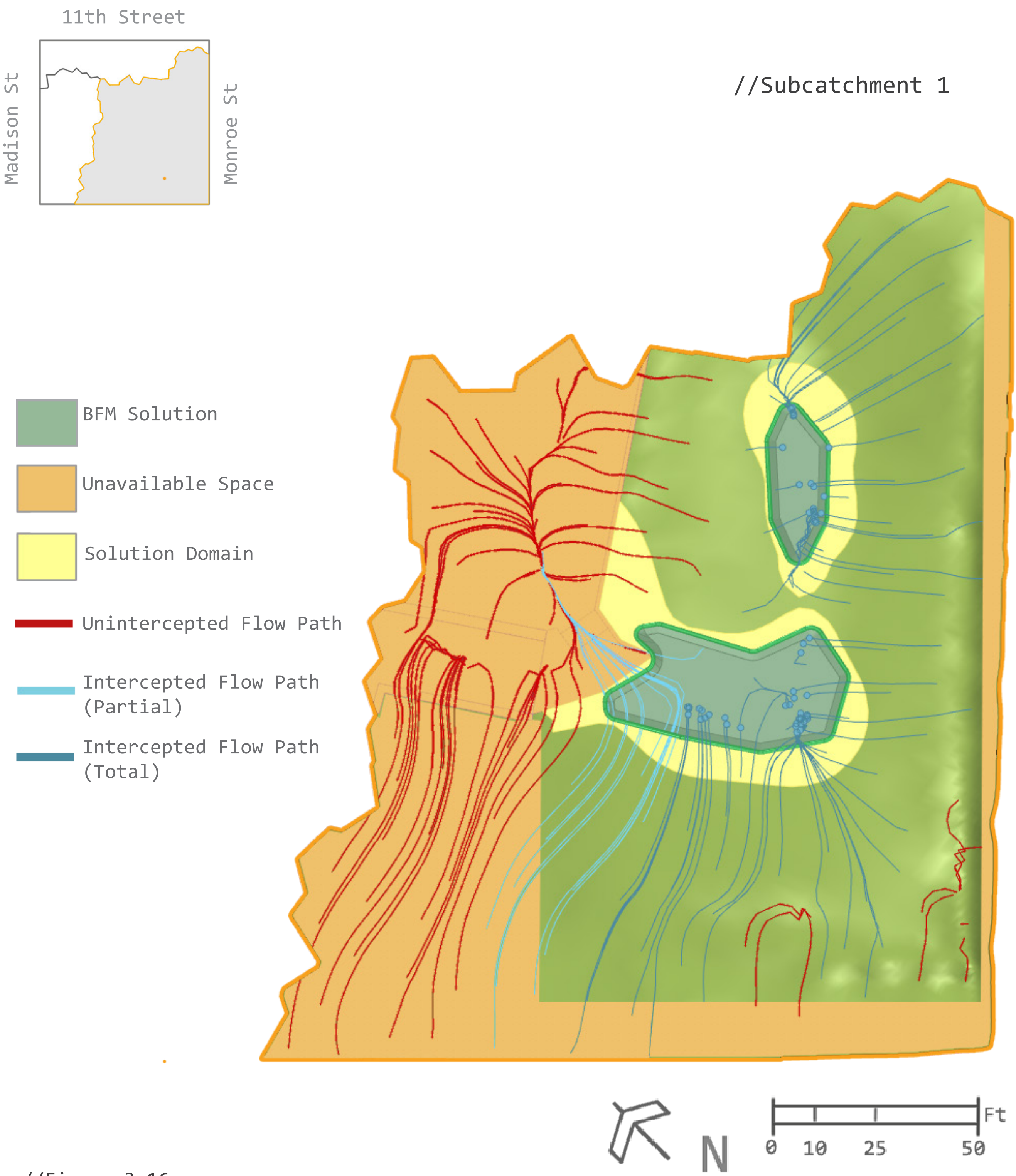
BFM solution: Despite the difficulty that the solution domain's shape posed, the elongated shape permitted BFM solutions that could reach out to intersect distant flow paths. This BFM is actually the only one of the four separate BFMs that performed in such a way. With each of the others, every flow path was captured at the end-point—meaning that the entire volume of that flow path was included. Altogether, BFM-1A captured 40 flow paths for their total volume and an additional 14 flow paths with partial volume. There are 6 flow paths within the solution domain that are not captured by the optimized BFM solution. As it happens, each of these missed ones were situated at the extremity of the protuberances on the solution domain. In order to include these outliers, the BFM solution would have to had been generated within an extremely narrow range of parameters for the vertex movement. Achieving that optimal solution was unlikely because Galapagos is a progressive solver. That is to say, it would have required a sequence of improbable solution iterations before the BFM shape could reach those distant flow paths. This type of shortcoming was unique to this BFM. None of the other BFMs failed to extend to flow paths within their solution domains.

Subcatchment 1: BFM-1B

Solution domain: The solution domain for BFM-1B formed around a cluster of flow paths terminating on the upper end of the terrace level. A subtle, trench-like depression in the terrain caused a linear shape to the domain. Conveniently, there were no constraints of any kind, allowing the full extent of the domain to stand.

BFM solution: A total of 29 flow paths were captured by the BFM solution, which accounts for all that were in the solution domain. Moreover, the runoff reduction obtained precisely matches the amount of proportional flow from those flow paths. In other words, the size—and therefore cost—of the solution do not exceed what is necessary to obtain total infiltration within the immediate context. This BFM demonstrates that, under the right conditions,

//Subcatchment 1



//Figure 3.16

Plan view of generated BFM solutions, solution domains, and affected hydrology.

the algorithm is capable of achieving ideal cost-effectiveness.

Subcatchment 2: BFM-2A

Solution domain: The solution domain for BFM-2A was highly constrained by the adjacent slope and pavement. As a result, its shape was relatively narrow and linear. Altogether there were 55 flow paths converging on this domain. Many of those originate on the terrace level or the surrounding slope, and are channeled below at the end of the retaining wall.

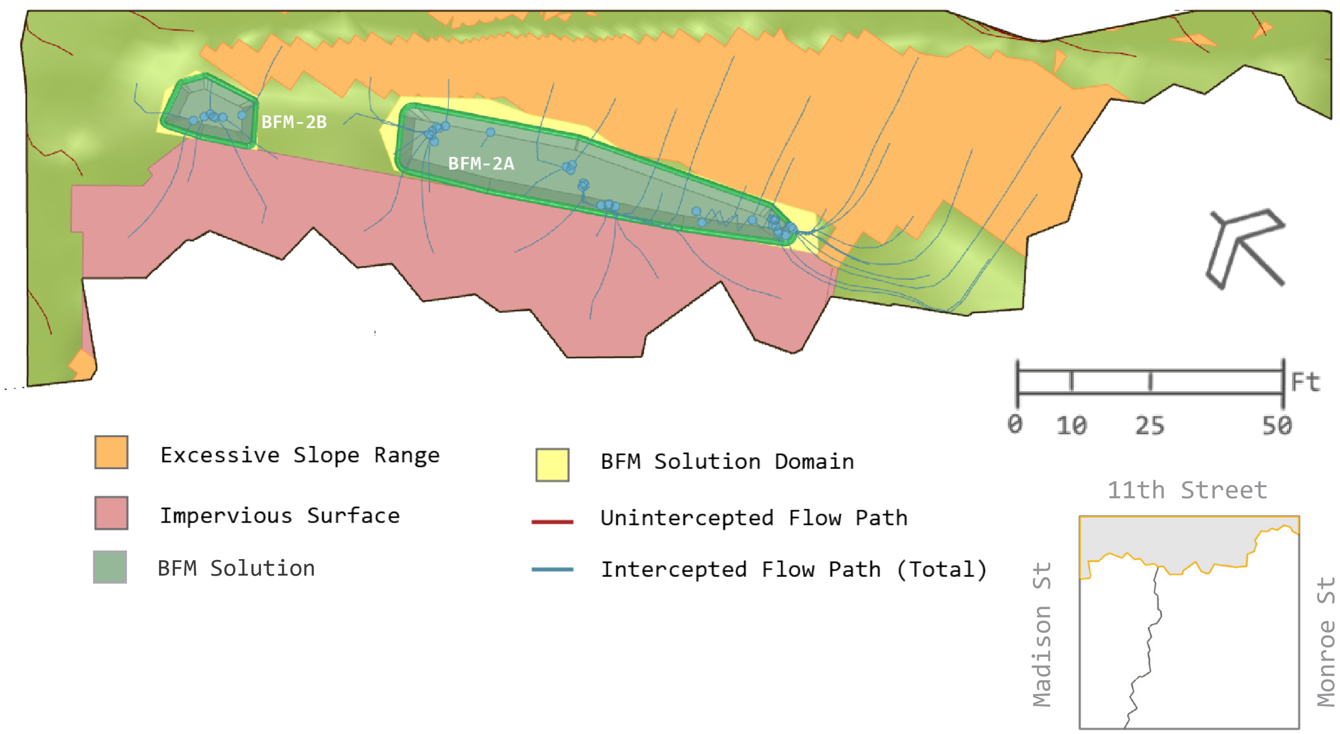
BFM solution: Despite the highly constrained solution domain, this BFM solution fully captures each of the 55 flow paths going into it. However, it falls short of total runoff reduction by 66 ft^3. It is the only solution that produced dimensions that were insufficient for the runoff target.

Subcatchment 2: BFM-2B

Solution domain: Here also, the solution domain was constrained to a narrow shape by excessive slopes and pavement. Relative to the others, this BFM was small with only 13 flow paths entering its confines.

BFM solution: This BFM solution was able to achieve total runoff reduction with very little solution space remaining. While sufficient in terms of performance, the BFM was impractically shaped. An obvious alternative would be an elongated shape that possibly connects with BFM-2A. This flaw is discussed further in upcoming sections.

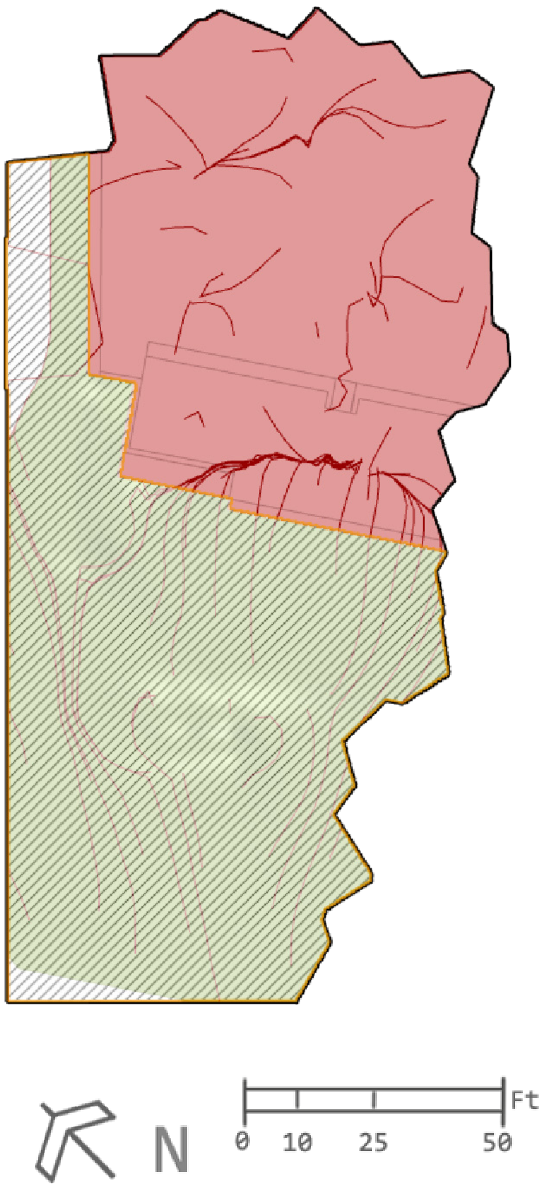
//Subcatchment 2



//Figure 3.17

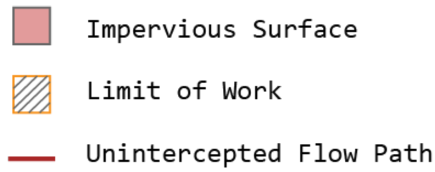
Plan view of generated BFM solutions, solution domains, and affected hydrology in Subcatchment 2.

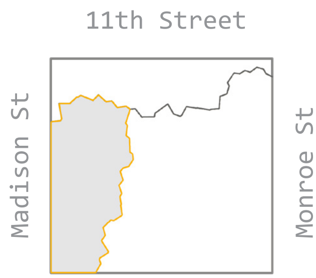
//Subcatchment 3



Subcatchment 3

No stormwater interventions were possible in this subcatchment as there was absolutely no space available in which to site them. Much of this subcatchment is situated outside of the limit of work and the space that is actually within the site boundary is entirely paved. Consequently, there is no reduction of the 2,386 ft^3 of runoff originating from this catchment —which drastically impacts the site-wide performance.





//Figure 3.18

Plan view of Subcatchment 3 with no generated BFM solutions.

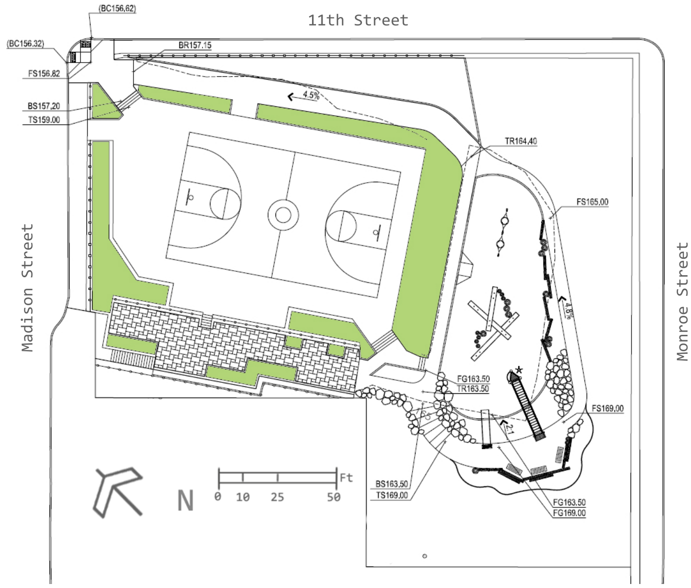
ii. GreenWorks Design

Conceptual Design Options

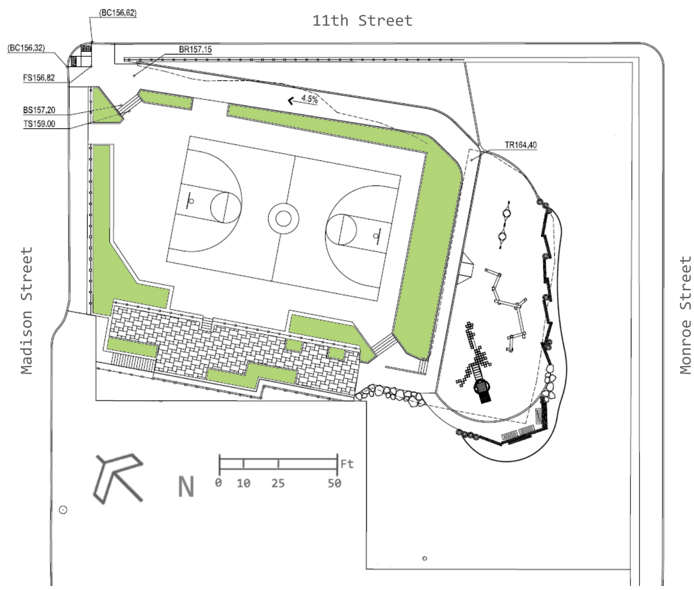
GreenWorks started their design of Latourette Park with two basic conceptual plans (Figure 3.19) which were largely based on the layout and program put forth in the city's concept plan (Refer to Figure 1.10, pg 26). Both options included the same fixed elements as the algorithmic solution—those being the paved sports court and the adjacent platform—and mainly differed in the proposed configuration of pathways and a new natural play area on the upper terrace.

Stormwater infrastructure followed the basic scheme proposed in the concept plan. The main facility was to be a large "bioswale" situated on the terrace level against the retaining wall. Additional swales and planters would then be dotted around the perimeter of the sports court. The most significant of these would be two infiltration planters wedged between the sports court and a proposed ADA ramp extending from the entrance to the terrace level. These initial concepts also featured planters on the platform, against the retaining wall where they would capture runoff from the parcel in the west corner. These initial concepts made a robust effort at trading off impervious area for vegetated stormwater facilities. And although stormwater facility placement was seemingly secondary to the placement of pathways and amenities, the extensive coverage was enough to significantly mitigate runoff.

//Option 1



//Option 2



//Figure 3.19

Conceptual design options for site-wide design. Proposed stormwater infrastructure is highlighted in green. (Adapted from Halpin, Margo and Johnson, Ben. "Latourette Park 30% Design," Sheet L 1.0, 2019.)

Preliminary Construction Documentation

Unfortunately, the robustness of the initial proposals had to be significantly downgraded in later design iterations for budgetary reasons. See Figure 3.20 for an adapted render of the latter design option. New constraints were added that extended the limit-of-work to exclude the platform. Thus, the stormwater management scheme would no longer directly intercept and attenuate the drainage from the property above. Additional items on the chopping block were the planters on the perimeter of the sports court. The full extent of the paved area would remain, with the exception of a 95 ft^2 infiltration planter near the entrance.

To offset the loss of the perimeter planters, the angle between the ADA ramp and the sports court was widened in order to accommodate slightly larger infiltration planters in-between. All other aspects of the plan were largely left unaltered. The terrace level would still feature the large bioswale—referred to as a "dry creek bed"—and natural play area. Also, soft-surface and gravel pathways were added around the sides of the play area and onto the adjacent steep slope.

Dry Creek Bed

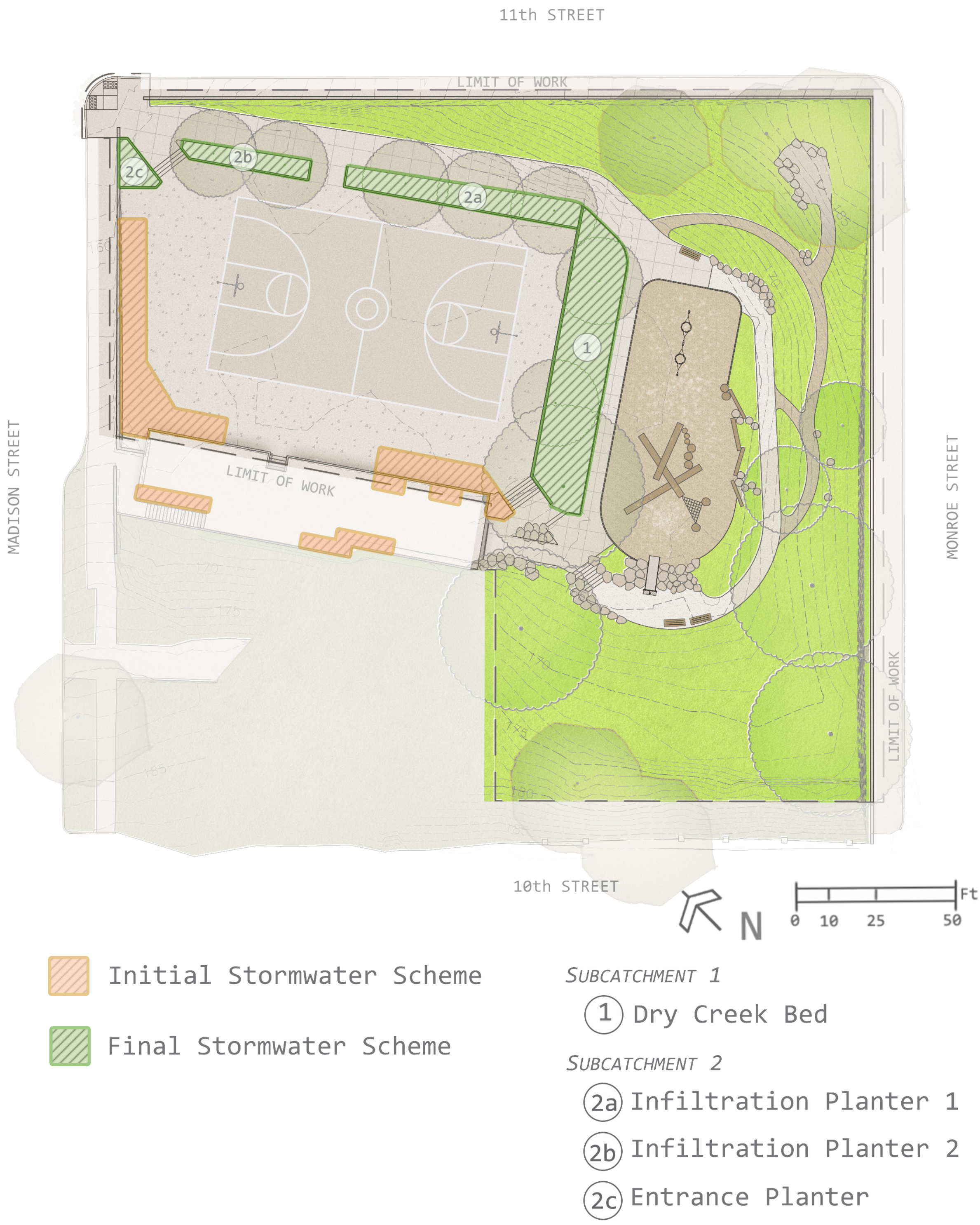
The dry creek bed design proposed by GreenWorks is very similar to that of the typical BFM represented in the algorithmic solution. Namely, it observes the same BMPs for dimensioning and material specifications (Figure 3.21). Those include sidewall slopes not exceeding 3:1, a maximum freeboard depth of 9 inches, a filter medium depth of 24 inches and a drainage layer depth of 12 inches. The forms do differ from each other in that the GreenWorks design is more aesthetically refined. True to the name of dry creek bed, the basin of this facility meanders in a curvilinear fashion in order to evoke the form of a natural channel.

The resulting basin area is 769 ft^2 and the total volume of the infiltration layers is 1,604 ft^3. This enables a runoff reduction of 1,320 ft^3 and assumes only a modest construction cost because of its simplicity. As the largest facility, it constitutes the main stormwater intervention of the GreenWorks design.

Infiltration Planters

There are three concrete-lined infiltration planters in the GreenWorks plan. Situated between the steeply sloped sides and the large paved area (Figure 3.22), these planters are sure to receive more inflow than they are capable of reducing by infiltration. A similar constraint was encountered in the algorithimic solution, which also fell short of total reduction. It may also be assumed that runoff would exceed the amount projected by the original hydrological analysis because of the addition of the ramp. The hillside would have to be cut to an even steeper angle and the ramp itself would add impervious surface to the subcatchment.

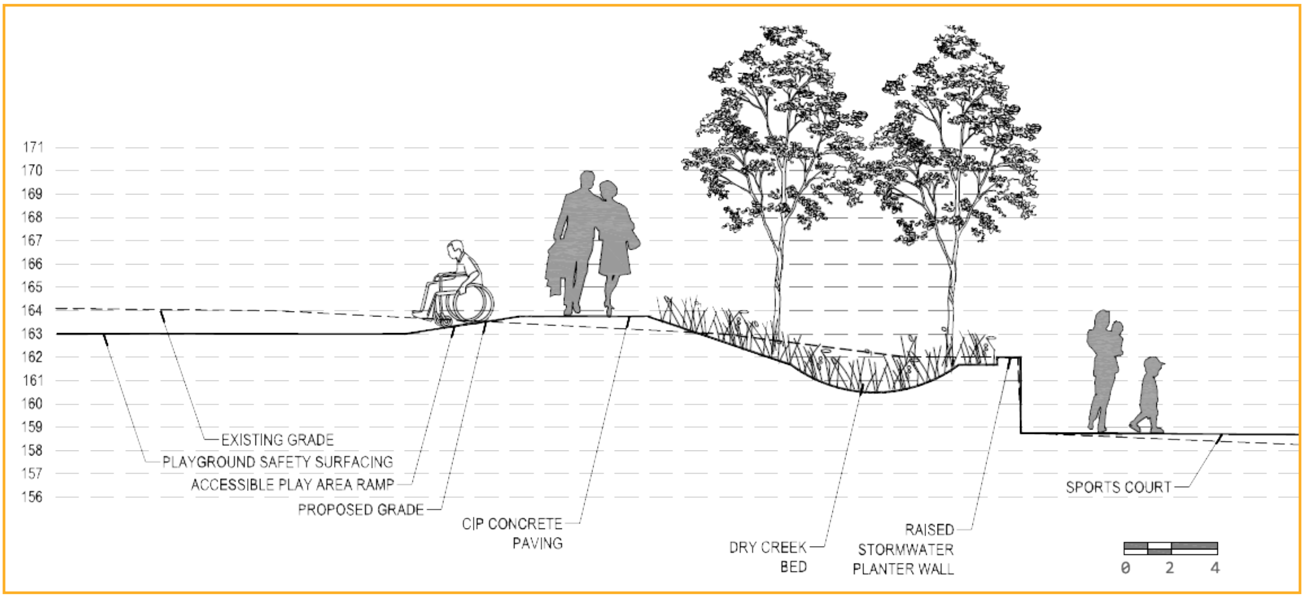
The planters' infiltration was calculated the same as that of bioretention facilities with the exemption of the area lost to the sloped sidewalls. Construction costs were considerably higher because of the difference in materials.



//Figure 3.20

Final concept plan for site-wide design with omissions from earlier stages highlighted. (Adapted from Halpin, Margo and Johnson, Ben. “Latourette Park 30% Design,” Sheet L 1.0, 2019.)

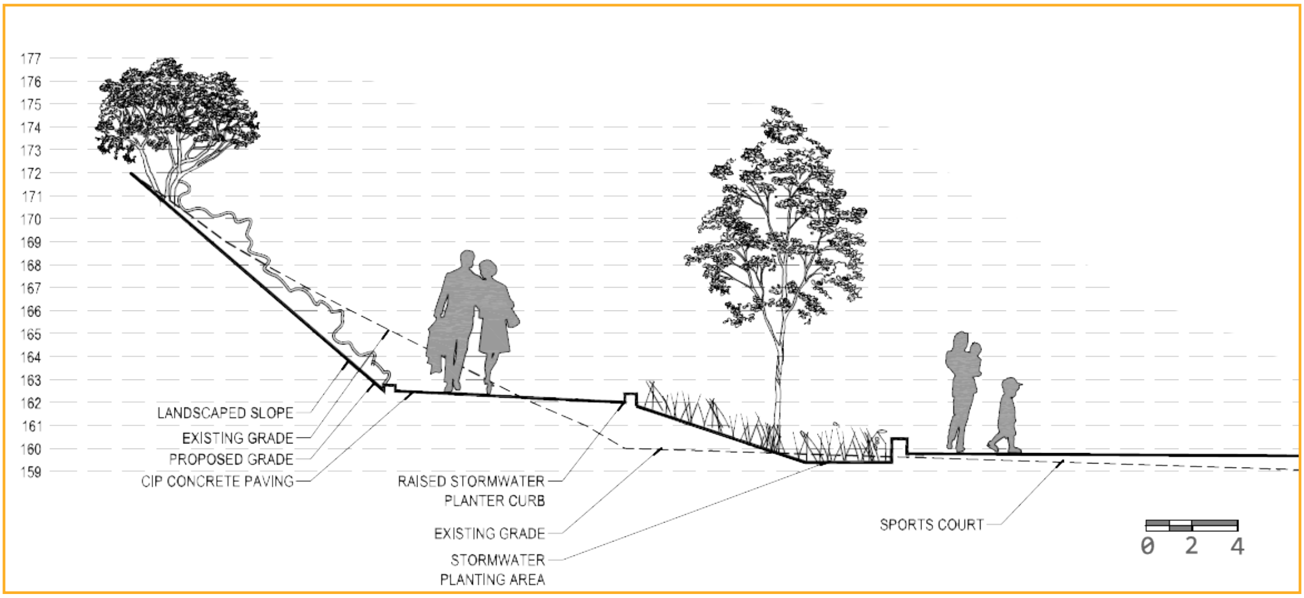
//Dry Creek Bed Section



//Figure 3.21

Section drawing of the dry creek bed stormwater facility and accessible play area. (Halpin, Margo. "Latourette Park 30% Design," Sheet L 2.1, 2019.)

//Infiltration Planter Section



//Figure 3.22

Section drawing of infiltration planters near the north entrance of Latourette Park. (Halpin, Margo. "Latourette Park 30% Design," Sheet L 2.1, 2019.)

iii. Evaluative Comparison

What follows is a side-by-side comparison of how the GreenWorks and algorithmic designs perform in terms of stormwater management. Each is evaluated for the amount and efficacy of runoff reduction as well as the associated costs of either design.

GreenWorks Runoff Reduction

Due to the additional impervious surface area added with the ADA ramp, the runoff volume would conceivably increase. Specifically, the runoff volume would change from 1,224 ft^3 to approximately 1,500 ft^3 in Subcatchment 2 following the change in percentage of impervious cover from 0% to 16%.

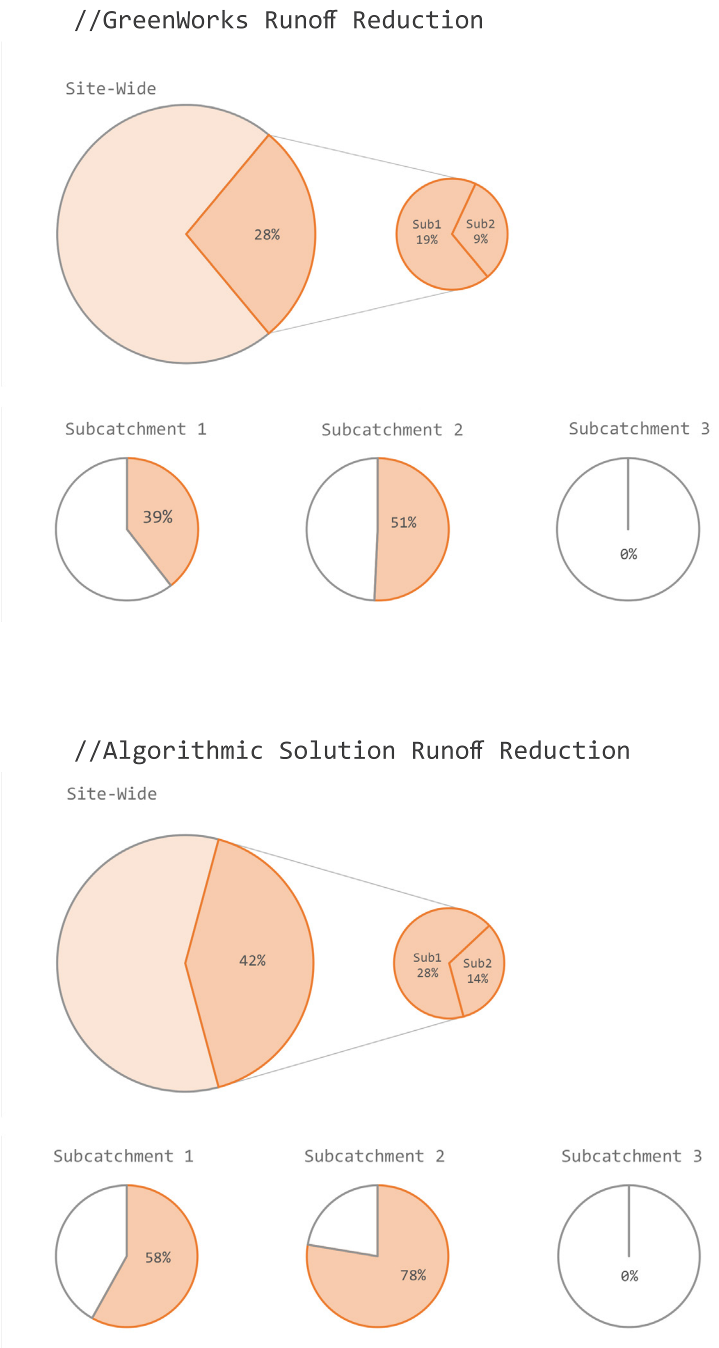
As it is, the GreenWorks performance in runoff reduction only represents the potential capacity of facilities and does not reflect how much runoff would actually be intercepted. Assuming that the hydrological model is correct, much of the surface flow would completely bypass the facilities. This is especially the case with the dry creek bed on the terrace level.

It could be assumed that much of the modeled surface flow would be intercepted by added elements—such as the playground or plant beds—or would flow down the paved path to the level below. Some of the paving added to the terrace level would work to the advantage of directing and intercepting surface flow. Such is the case on the northern end, where most of the surface flow coming off the hillside could be diverted into the stormwater facility. However, most of the rest of the runoff in this subcatchment would not reach the facility. The mulch-surfaced playground and its landscaped buffer would likely intercept most of the runoff extending from the eastern and southern edges. Additionally, a significant amount of runoff would flow down the path and stairs to the lower paved area.

In the absence of additional measures, such as runnels channeling distant flow into the dry creek bed, the placement of this facility would not perform to its full capacity. The other three stormwater facilities would not have the same problem. A slight cross-slope on the ADA ramp would ensure runoff inputs to the infiltration planters and the entrance planter would presumably receive a high runoff input from the surrounding pavement. More than likely, these facilities would receive significantly more runoff than they would be capable of infiltrating. Added together, they reduce runoff reduction via infiltration by 621 ft^3—a decent proportion of the approximate 1,500 ft^3 of runoff from Subcatchment 2.

Where this design suffers most is in Subcatchment 3. There were no interventions made here due to the fact that there was no open space available. The initial design options would have accommodated a significant runoff reduction with the added planters. But without those, the entirety of the runoff would be discharged from the site via storm grates or as sheet flow.

Taking into account the potential capacity for runoff reduction only, altogether the GreenWorks design accomplishes a site-wide runoff reduction via infiltration of 28% (Figure 3.23).



//Figure 3.23

Runoff reduction attained by either design with breakdown across subcatchments.

Algorithm Solution Runoff Reduction

The algorithm stormwater solution ostensibly performs better than the GreenWorks design—albeit before design considerations of where to place the added features of the ADA ramp and the nature play area. Where it was possible, the algorithmic solution found efficient surface shape configurations for targeted interception. However—as was the case with the GreenWorks design—Subcatchment 3 presented no opportunities for stormwater interventions without altering the paved surfaces. Consequently, site-wide performance for runoff reduction was significantly impacted. Isolated from site-wide performance, Subcatchments 1 and 2 managed a runoff reduction of 58% and 78%, respectively.

For Subcatchment 1, much of the runoff was directly intercepted by the two BFMs. Each of those were sized precisely according to the proportional volume of runoff directly intercepted by the facility—thereby optimizing the ratio of runoff reduction to costs. Nearly all of the unintercepted runoff was impossible to address within the algorithm's constraints as it ran through space that was off-limits. The algorithm generated solution domains based on flow path end-points—so these flow paths were missed because their end-points fell well within the off-limits space on the pavement below.

The solution for Subcatchment 2 generated a very similar layout to that of the GreenWorks design, with linear facilities abutting the edge of the pavement. The larger of the two BFMs was not able to completely reduce its proportional amount of runoff. Of the 856 ft^3 of runoff input, the BFM solution was only able to infiltrate 794 ft^3. This is because its solution domain was greatly constrained between steep slopes on one side and pavement on the other. Curiously, the solution domain did not give enough space for the BFM to expand laterally. That would be the common sense solution that the algorithm missed.

All told, the site-wide runoff reduction was 42%—a substantial increase over the 28% achieved without the algorithmic approach. Still, the goal was for a 100% reduction through infiltration. With this solution, a substantial amount of runoff is still being discharged from the site untreated.

GreenWorks Costs

The GreenWorks plan incorporated both infiltration planters and a bioretention facility—priced at $50/$ft^2$ and $15/$ft^2$, respectively. The dry creek bed is—for all intents and purposes—a bioretention facility with a simple construction. At 769 ft^2 this facility totaled to $11,535 in construction costs.

Altogether the infiltration planters totaled to 551 ft^2. Despite having a combined area significantly lower than the area of the dry creek bed, these would end up costing a total of $27,550. Although, given the conditions created by the addition of the ADA ramp, there are no reasonable alternatives to the type of facility to fit the narrow, inclined space.

Site-wide maintenance—calculated at a flat rate of $0.25/$ft^2$ annually over a 30-year period—came out to $9,900. When factoring overall cost against the site-wide runoff reduction rate, it was found that the cost-effectiveness of the GreenWorks plan is $1,749 per 1% reduction. Figure 3.24 breaks down the cost-runoff relationship across the separate facilities.

Algorithmic Solution Costs

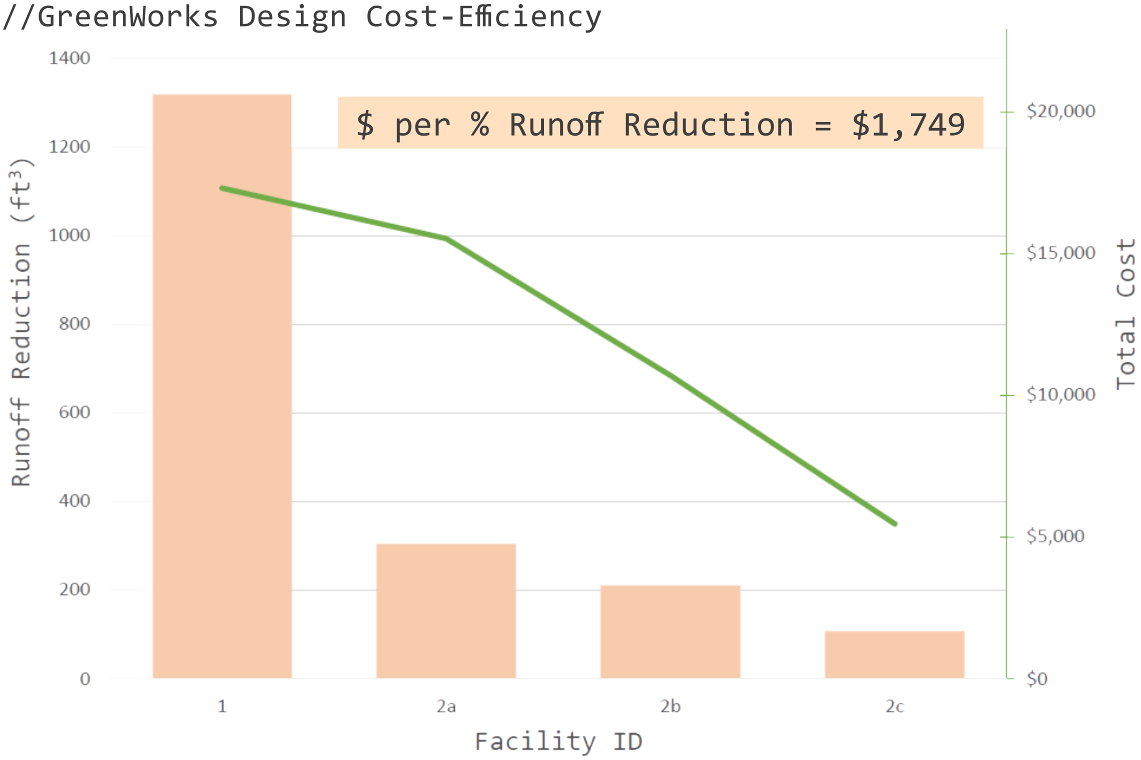
The algorithmic solution was limited in stormwater component options to just bioretention facilities, and therefore would not incur the high cost of construction that accompany the infiltration planters. The area of all the facilities combined totals to 1,436ft^2. At a rate of $15/$ft^2$, this would result in construction costs of $24,285.

Maintenance costs for the algorithmic solution are factored the same as with the GreenWorks plan. Calculated at a rate of $0.25/$ft^2$ annually over a 30-year period, it would total to $12,143. Added together, the total cost of this solution is $36,428. In terms of cost-effectiveness, it equates to a rate of $867 per 1% reduction. Refer to the lower chart in Figure 3.24 for a summary of the costs of individual facilities within the solution.

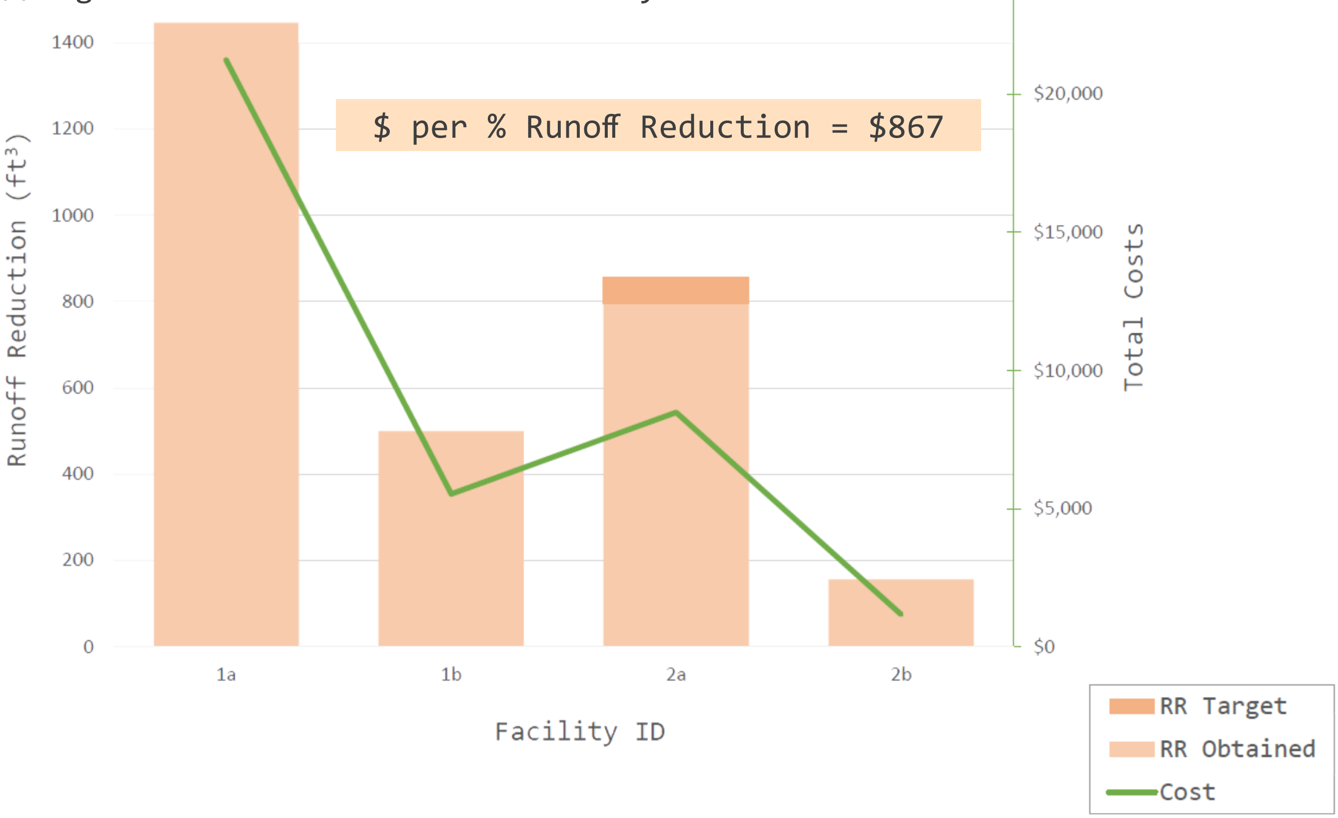
//Figure 3.24

Costs-efficiency of either design. For each facility, the figure displays total costs related to runoff reduction.

//GreenWorks Design Cost-Efficiency



//Algorithmic Solution Cost-Efficiency



3.4//Preliminary Site Design Recommendations

The result of the algorithmic solution puts forth a design that is only partially developed. To truly be on equal footing with the GreenWorks site plan, the design stemming from the algorithmic solution needs to make considerations for additional features which are beyond the limitations of the algorithm.

The following section is the direct design response to the algorithmic solution in which several essential elements from the city's concept plan are integrated into the site design. In doing so, it is determined whether conflicts arise between the solution and the overall intent of the park's renovation. Additionally, proposals are made to enhance results for a follow-up execution of the algorithm. These changes enable a new iteration of the algorithm with changes updated to solution constraints.

i. Added Elements

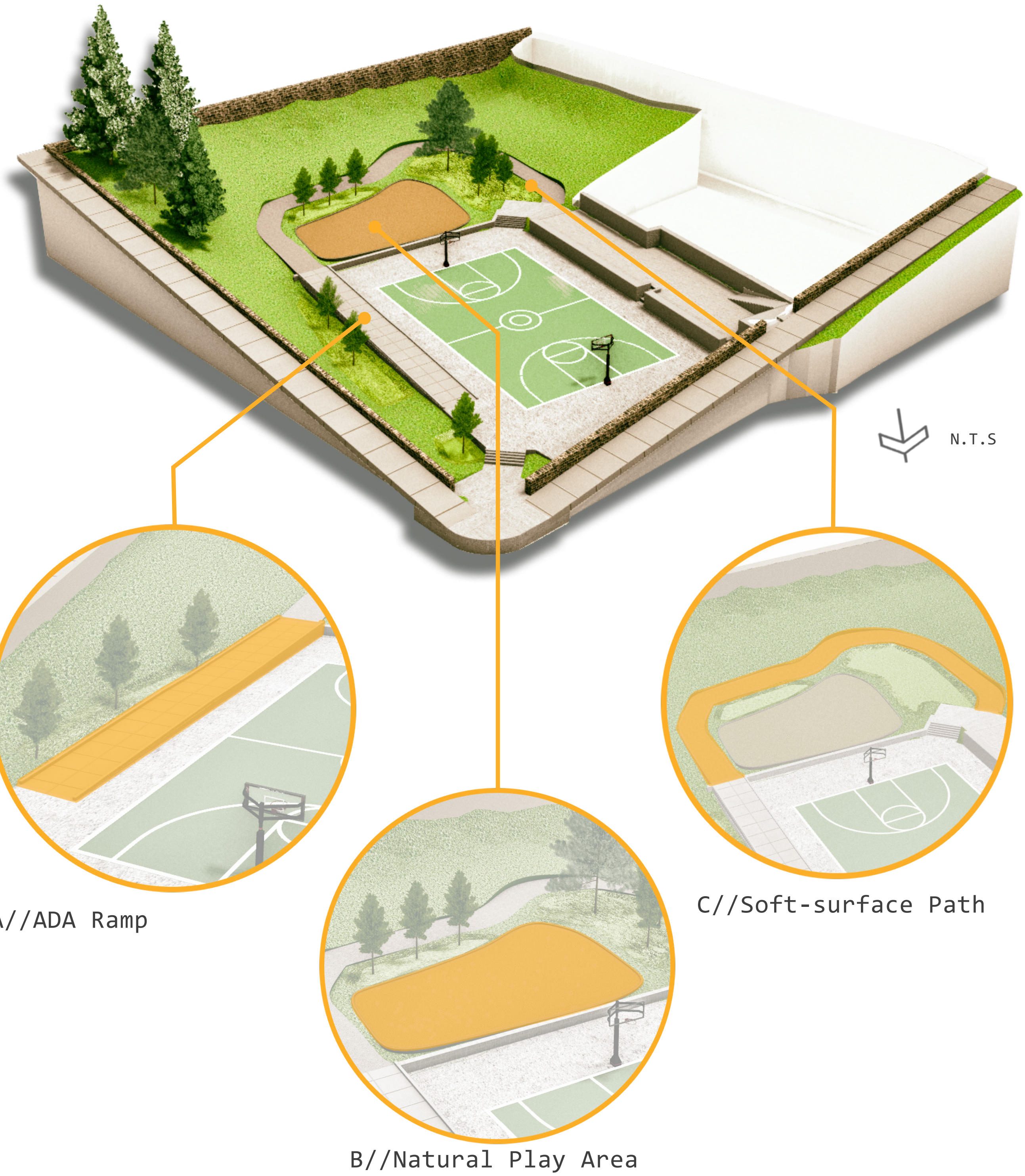
Of the key elements included in the client's initial concept of a revitalized park space, the glaring omissions following the algorithmic solution were improvements to circulation and the addition of a natural play area. Each of these features were added to the site without altering the intial placement of the four BFM solutions.

Circulation improvements included both an ADA ramp between the paved and terrace levels and a soft-surface path that extends from the top of ramp. The ADA ramp (Figure 3.25, part A) was added within a space that was otherwise slated to remain impervious on the northern edge of the sports court. In comparison, the GreenWorks plan includes an added ramp that runs parallel to the edge of the sports court with a buffer of planters in-between. Doing so necessitates a slight regrading, and also increases the footprint of impervious surfaces on-site. By shifting the position of the ramp, more space is retained for vegetation and stormwater facilities—and without any regrading in the process. The drawback of this choice is that the open space on the terrace above is slightly more constrained.

That condition impacts the placement of the natural play area (Figure 3.25, part B). The BFMs in Subcatchment 1 and retaining wall also confine the space so that the available area is essentially defined on four sides. The resultant play area is 1,962 ft^2—an area that is approximately 25% smaller than the GreenWorks plan. Another key distinction between the two is the relative position of the play area. In the GreenWorks plan, the dry creekbed separates the playground from the retaining wall—effectively buffering the playground from the level below.

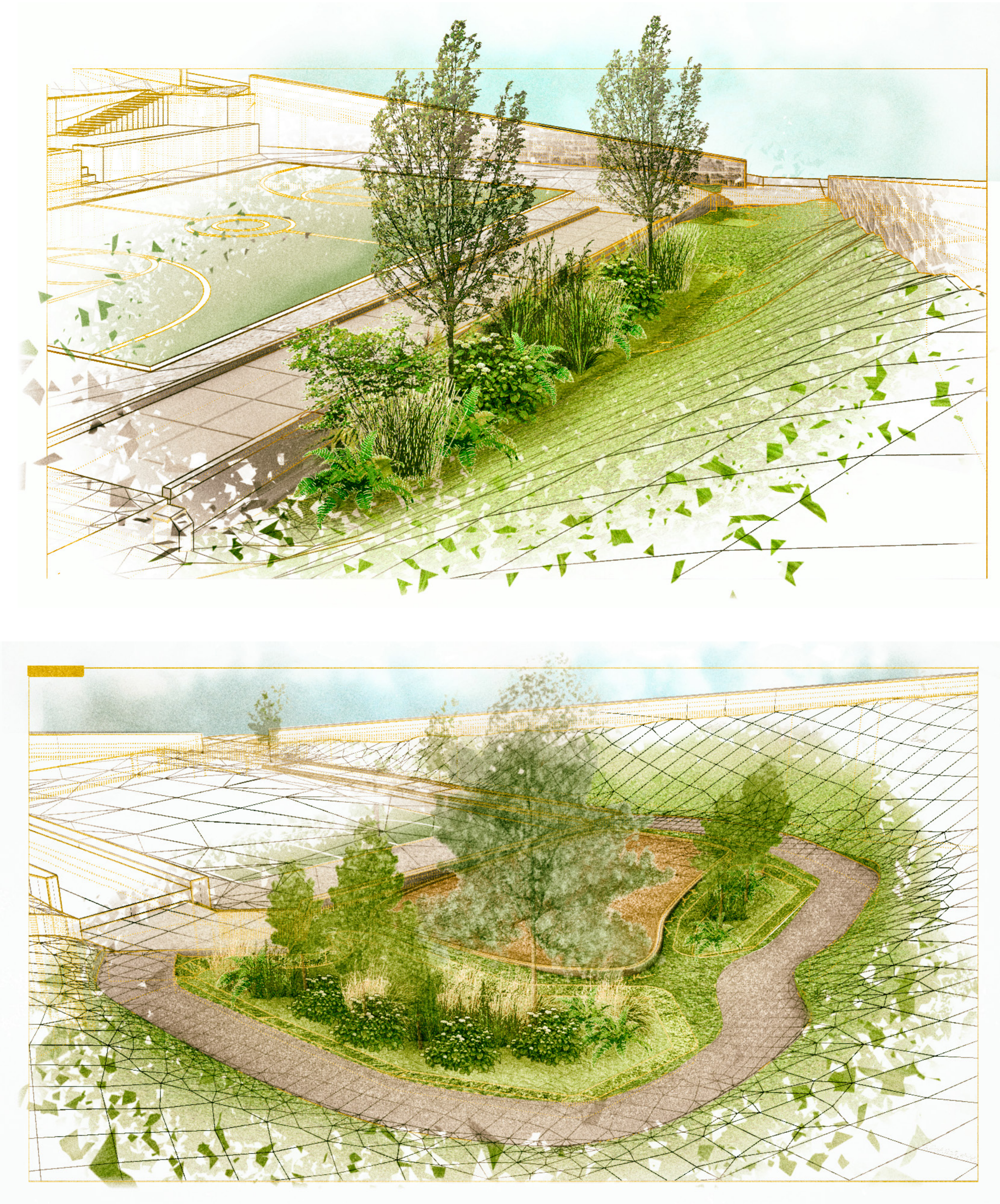
The soft-surface path (Figure 3.25, part C) then wraps around the play area and BFMs, connecting the ramp with the hard-surface landing above the stairs on the southern corner of the sports court. The position of the path, coupled with the gap between the two BFMs, creates an opportunity for a secondary entrance to the play area.

Rendered depictions of these added elements in the site design are presented in Figure 3.26.



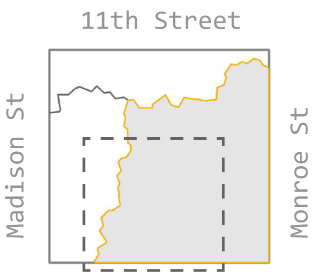
//Figure 3.25

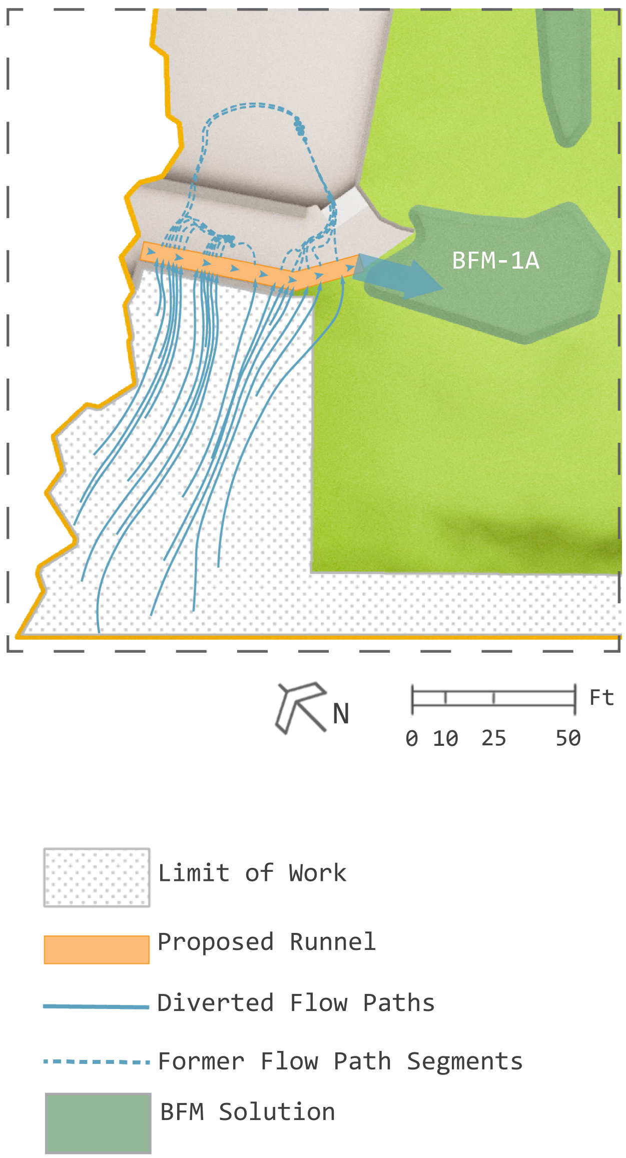
Added features to the site design following the completion of the algorithmic solution for storm-water design. Part A highlights the ADA ramp, part B highlights the natural play area added on the terrace level and part C features the soft surface path connecting the previous features.



//Figure 3.26

Preliminary design rendering of Subcatchment 1 (below) featuring added play area and path, and (above) Subcatchment 2 with added ADA ramp.





//Figure 3.27

Revised stormwater interventions for Subcatchment 1.

ii. Model Revisions

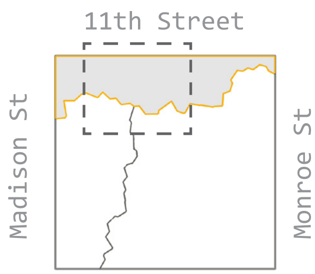
Beyond adding desired site features, the purpose of this basic design response is to reparametrize the algorithm. Ahead, each part of the algorithmic solution is assessed for changes that that may refine results in subsequent executions of the algorithm. Any changes enacted through these proposals would fundamentally alter the model which, in turn, may produce different solutions. Ideally, this process could repeat until an agreeable solution is generated.

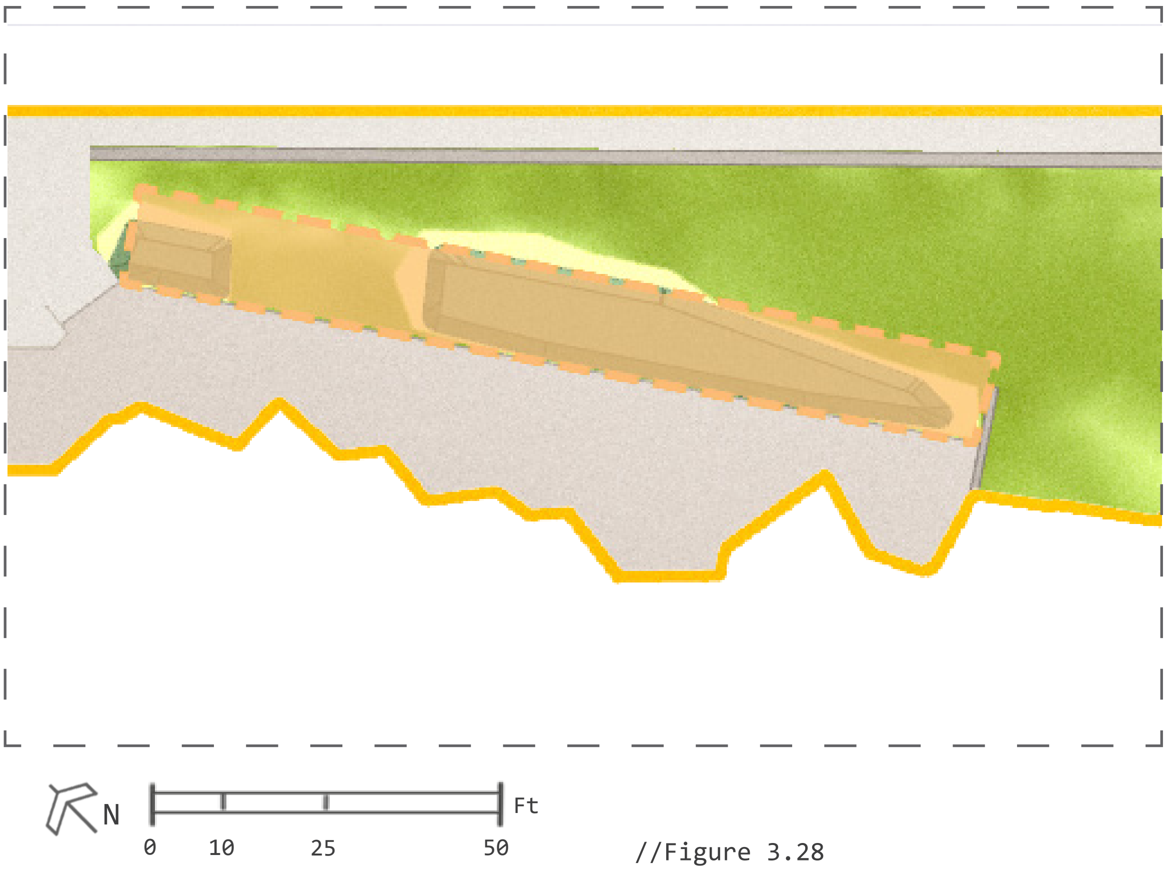
Subcatchment 1

Generally, Subcatchment 1 performs very well—with the exception of not being able to intercept the surface flow originating in the off-limits space on the western side of the catchment. All of that flow might be captured with the addition of a simple intervention. A grate and/or runnel may be installed at the edge of the concrete platform and channel diverted runoff into the adjacent BFM (Figure 3.27). With this proposed intervention, the localized hydrology would need to be remodeled to include the substantial increase in runoff reaching the BFM. Subsequent executions of the algorithm would then likely generate larger solution domains and larger BFM surface shapes.

Subcatchment 2

The two BFM solutions of Subcatchment 2 perform well in terms of runoff reduction, but are impractical because of their shape. As they are, the solution domains and the BFM solutions were tightly constrained by slope conditions and impervious surfaces. This resulted in them having a predominantly linear orientation along the northern edge of the sports court. A problem exists in that there is a gap of approximately 30 ft between them. In a realistic design scenario, a preferred solution would combine the two into a single linear basin. To accomplish this, the solution domains would need to be expanded using a multiplier, and then the algorithm could configure the BFM solution accordingly (Figure 3.28). This revision expands the available space for BFMs, and could accompany additional interventions in Subcatchmet 3.



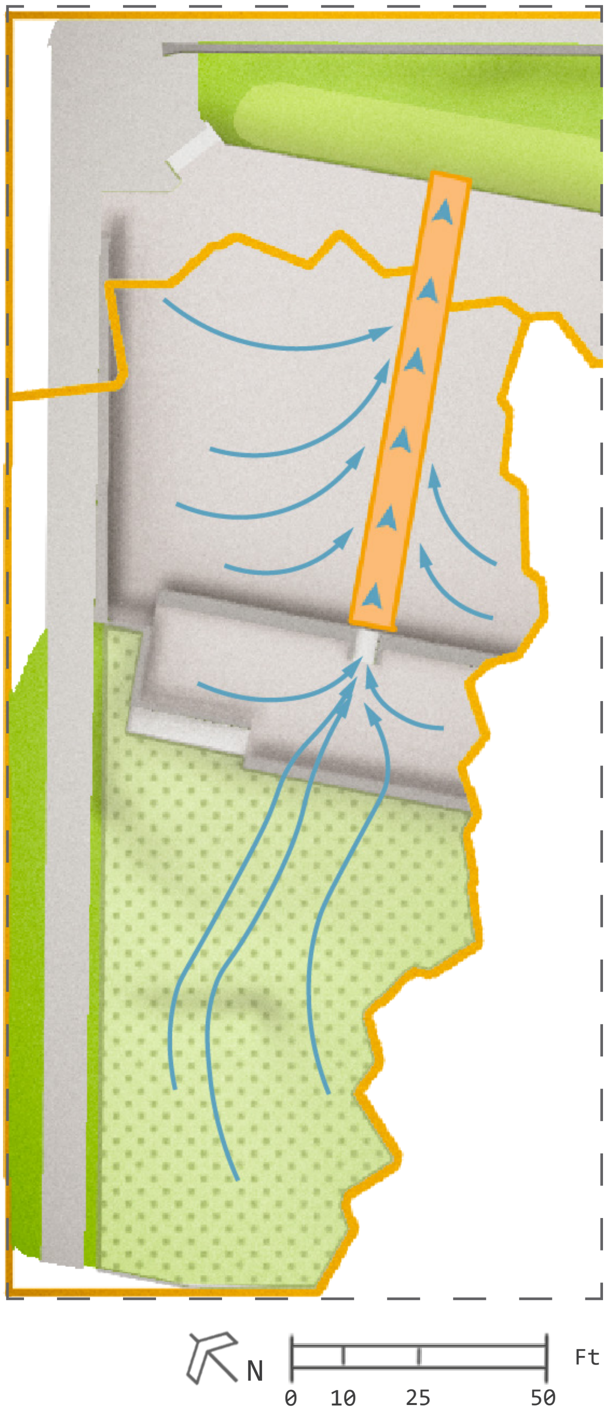


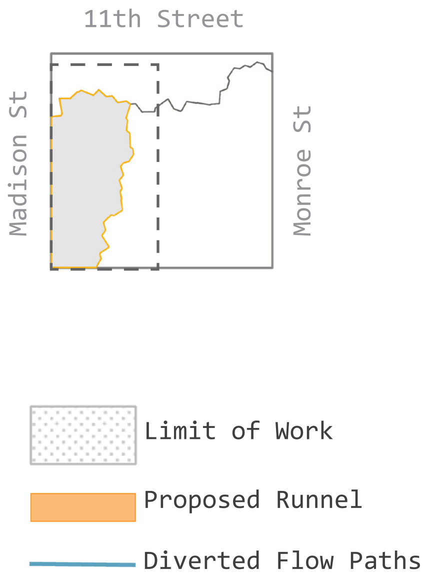
//Figure 3.28

Revised stormwater interventions for Subcatchment 2—combines BFMs.

Subcatchment 3

Subcatchment 3 presents the most challenging issue to the algorithmic approach. Because the algorithm is programmed to localize solutions and there was absolutely no space available within the subcatchment, no solutions were provided in the first iteration. In order to use the algorithm for a solution to this dilemma, the runoff needs to be directed to Subcatchment 2 where there is additional space available. Drains would need to be installed under the pavement and connected to the other catchment. In a subsequent execution of the algorithm, the hydrological simulation would effectively shift the target reduction volume to Subcatchment 2, and proceed to automatically generate a better solution.





//Figure 3.29

Revised stormwater interventions for Subcatchment 3—diverts runoff to Subcatchment 2.

"As researchers, we operate under the assumption that all models are wrong—and some are useful."

— Alex Felson[1]

1 Felson, Alex. "The Role of Models to Inform Green Infrastructure." Lecture at Simulating Natures Symposium, Philadelphia, PA, March 2015

4 Conclusion

4.1//Refinements to the Project Approach

i. Limitations

The focus of this project was to take a novel approach to a stormwater design scenario with the principles and tools of algorithmic modeling. In doing so, the expectation was to develop the means of generating optimized solutions for siting and sizing bioretention facilities on a site-scale. Rhino+Grasshopper was used to produce the algorithmic model with that capability, and results were evaluated against a professional design. Results were ostensibly favorable, but a more in-depth examination of the approach is warranted. At the start of this project, I asked the question:

> *Can a computational design approach be used to algorithmically generate site-scale stormwater solutions that are optimized to site specific constraints and performance targets?*

The short answer to this question is—yes, but with a caveat. The logical workflow of the algorithm tracks with standardized methodologies of assessing and responding to stormwater conditions as presented by the SuDS Manual. Furthermore, the algorithm actually works. It generates optimized solutions without runtime errors. That in itself is an encouraging outcome—but being able to run without breaking is the low bar for success. Another way to approach this question is in asking: does the algorithm generate valid solutions? To that I answer that the results serve well as approximations of site-scale stormwater designs but could certainly be refined further.

Hydrological Simulation

The algorithmic solutions are predicated on a number of assumptions. The first of which is that the hydrological simulation yields useful results. I have no doubt that the digital simulation produces the same outcome as the same process would with pen and paper, but faster. However, I believe there is more potential in using the digital method other than providing speed. It could paint a fuller picture of the site hydrology.

I believe it could do this through an enhancement of both the calculation and depiction of surface flow. In reality, the nature of runoff is to move across the terrain in sheet flows. But that quality could not be accurately represented in the digital simulation. Instead, dynamics of flow were simplified and abstracted—its areal qualities represented by lines and points. The generated flow paths served as approximations of the direction of sheet flow in the space around the flow path. To be sure, no better alternative to this simulation method was found in the course of this project. However, it may be argued that the same method could be used with greater precision. Some 200 flow paths were randomly generated over an 0.8-acre area, but there is conceivably no upward limit to that number. An enhanced use of the simulation would increase the number of flow paths until they start overlapping each other.

So why was the model limited when there was potential for improvement? It was a choice in regards to computational processing speed and the practical execution of this project. Any higher level of precision would have been too demanding on my computer hardware. In no part of the algorithm was this more of a concern than with the hydrological simulation. On top of limiting the number of flow paths, the underlying mesh had to be simplified for the algorithm to be operable. Initial attempts were made using a terrain mesh with approximately 10 times the resolution than that of the selected mesh. I had to systematically test a series of meshes of decreasing resolution before finding the appropriate one. The takeaway is that more processing power would enable better accuracy in simulating site hydrology, and therefore more accurate results for dependent calculations such as targeted intervention and runoff reduction.

The hydrological simulation could be conceptually improved by adding an account of flow velocity. This is an important parameter to more detailed designs of stormwater facilities—especially at a site like Latourette Park with its extreme slope range. Excessive inflow velocities create upkeep issues by scouring the sides of the facility basin and depositing unsustainable quantities of sediment. There are design responses to this problem but first the excessive condition needs to be identified. An additional capability may be added to the algorithm that measures velocity at points along a given flow path using Manning's kinematic method of measuring sheet flow:[2]

$$T_t = \frac{0.007\,(nL)^{0.8}}{(P_2)^{0.5} S^{0.4}}$$

where:

T_t = travel time (hr)
n = Manning's roughness coefficient
L = flow length (ft)
P_2 = 2-year, 24-hr rainfall (in)
S = average slope (ft/ft)

2 "Sheet Flow References," (NRCS National Water and Climate Center website. Last modified July 7th, 2011, https://www.wcc.nrcs.usda.gov/ftpref/wntsc/H&H/WinTR20/SheetFlowRefs.doc).

Points that exceed a specified threshold may be highlighted and then factored into the procedure for constraining the solution domain. Alternatively, the information would be useful in determining whether grading changes or erosion control measures would be warranted in the broader stormwater management scheme.

Another assumption made is that the runoff reduction calculations are correct and comprehensive. However, one aspect of runoff reduction that was not covered in this approach was the benefits afforded by vegetation. Plants reduce runoff through evapotranspiration—a process in which water is absorbed or intercepted by the plant and then conveyed into the atmosphere. Some research has concluded that mature vegetation can reduce runoff by as much as 80% compared to asphalt in the same environment.[3] While this is understood as a significant factor, I was unsuccessful in incorporating a method to calculate quantities of evapotranspiration. Therefore, the final tallies for runoff reduction are limited to the amount reduced by infiltration and the overall reduction is presumably much higher.

Solution Variablility

At the onset of this project I estimated that algorithmic modeling would provide an extensive variety of design options—at least in terms of multiple unique BFM shapes within each solution domain. I fully expected to demonstrate that fact and assert it as an improvement over conventional design. However, outcomes did not match expectations and there was a notable lack of variety in both the shape and placement of the algorithmic solutions. Theoretically the evolutionary-solver should have produced different results with each run-through because the variations between generations have a random element to them. Instead I found that each run-through produced nearly identical results or became stuck in a *local optimum*.[4] This tells me that the solutions to this scenario are intrinsically more deterministic that I originally postulated.

Alternatively, it may be a matter of how the approach was executed. Perhaps the problem formulation in the representation phase was overly-constrained, offering only a narrow range of near-ideal solution options. To test whether that is the case, it would be worth experimenting with a wider range of dimensioning parameters for the BFM. Such variables include the depths of the freeboard and sub-grade layers, or slope of basin sidewalls. In the execution of this project, those values were locked-in at their maximum values with the assumption that greater volume corresponds with higher runoff reduction. Regardless of whether this assumption is correct, an adjustable value range may produce a wider variety of shapes and spatial configurations.

Up to this point, the concerns over the model's validity have all regarded the level of accuracy of its methods and measurements. A different way of approaching the question of validity would be to consider how the results would be received in the

3 Woods-Ballard et al., *SuDS Manual*, 363.

4 A *local optimum* is a solution produced in an optimization problem that is better than the neighboring solutions. This is opposed to a global solution, which is best among all possible solutions.

overall site design process. That concern is addressed in the second research question:

> *Can the design solutions generated by this approach be feasibly incorporated into a real-world site design scenario?*

Again, the answer is yes—but with conditions. The algorithm was very effective at optimizing solutions within the constraints presented in the problem. Given the particular scenario of siting and sizing a single type of facility within the proposed spatial limitations, the algorithm generated solutions for the cheapest possible configurations to achieve a very specific runoff reduction target. The dilemma is that these are very narrow results for what is often a complex problem. Consider the implications of such a confined solution.

Firstly, bioretention facilities are one of many types of infrastructure that may work together within a site-wide stormwater management system. A better solution for this site might have included pervious paving, swales and/or filter strips. Unfortunately, other types of infrastructure were not accommodated in this project's approach for reasons of scope.

Another concern is the conflict that might arise between generated stormwater solutions and other aspects of site design. For instance, the placement of a bioretention facility may fundamentally interfere with the intent of the space around it. These are choices made at the designer's discretion. They are made through multiple iterations that factor all the various aspects of site design. It is well outside the capability of the algorithm to do the same.

ii. Model Revisions

Despite the short-comings listed in the previous section, I believe that the algorithmic modeling approach could be very valuable for finding cost-effective, performance-based solutions in landscape architecture. It would simply require a more rigorous execution of the framework—with multiple iterations of generated solutions. And with each iteration the designer would need to assess the solution outcome and progressively update the model accordingly.

Doing so in context of this project would go as follows: the first iteration of the model would determine where bioretention facilities are most effectively used in status quo conditions. Subsequent iterations could selectively incorporate solutions and then rerun the hydrological simulation for different results. Designers could also test interventions beyond the discrete addition of stormwater facilities, such as site-wide changes to surface materials or grading. It would be relatively simple to alter the terrain mesh to include a conveyance channel to link runoff to facilities. And changing impervious surfaces to pervious ones could be easily factored into the model by changing the associated coefficient of runoff. Other types of stormwater facilities could be modeled as well, but would require their own algorithmic definitions for constraints and objectives.

In this way, the approach developed in this project could be used for an ongoing stormwater analysis—supplementing the overall design process with changing information generated from a progressive modeling process. This would ensure that BMP goals are actually met, and potentially at the lowest cost.

4.2//Transferability

In addition to considering how this approach can be improved for the given scenario, there is also potential for it to be adapted to a variety of landscape design scenarios. As is, the algorithm can be applied to other stormwater conditions and at different sites. Users may do so without a deep knowledge of how the algorithmic process works, as this algorithm includes easily adjustable parameters and dropdown menus to choose between different design storms and BFM dimensions. Likewise, the base mesh may be easily swapped out to represent different sites for which terrain analysis functions are essentially automatic. Data sorting components then automatically cross-reference values obtained from terrain analysis with the user-set storm parameters. These values are then plugged into the hydrological simulation. This type of linkage continues on up to the start of the evolutionary solver—requiring minimal input/control from the user throughout the process.

In another sense, the project is transferable in that the algorithm can be directly expanded on. In ways already discussed, the algorithm can be improved for more robust results in this project's scenario. It may also be furthered and adapted to similar types of design problems that involve optimizing spatial configurations of landscape elements. The effort of that conversion would be relatively minimizal as the mechanics and basic structure of this algorithm could be applied.

The same approach may be taken to translate the concepts of another design scenario into a computational representation and search process—allowing for the same solution optimization process but with other objects in the landscape. One such design scenario would be optimizing planting arrangements for cost-effectiveness and a wide assortment of environmental benefits. Given the parameters for individual plants' spacing needs and growing conditions, the same method of solution optimization may be used to test different configurations of plants within a limited space. Using the same operations from this project, an adapted algorithm could automatically determine aspect, shading, moisture and soil properties—which could all be used as search parameters.

The objective parameters in this scenario may then be customized to the various quantifiable benefits of a planting arrangement, including: atmospheric pollution filtration, plant health, carbon sequestration, microclimatic cooling and oxygen production. Results may even be factored into the stormwater scenario if the evapotranspiration rates were assessed as well.

4.3//Reflections

I started this project with next to no knowledge of how to code algorithms. Predictably, the process of developing my own has been strenuous, frustrating and sometimes disappointing. Altogether I estimate that I spent over 300 hours in Grasshopper to get to the final outcome—and even then I feel that there are many ways that the algorithm can be improved. On multiple occasions I was forced to downgrade my expectations of the algorithm's functionality—having started with goals that were too ambitious and perhaps a bit too fuzzy.

Despite the difficulty, I would emphatically encourage any landscape architect to gain at least a modicum of coding ability. Like anything else, the process was slow to learn at first but eventually became easier to grasp. Once the skill is gained, there are a plethora of design applications that were not touched by the scope of this project.

Beyond considering time and effort, there is more to the question of what the personal price is to take on the algorithmic modeling approach. Of foremost concern, there is the issue of how the role of the designer changes in an approach that essentially automates certain aspects of design. This project does not address that concern directly, but having assumed the roles of designer and researcher, I do have my own conclusions on the matter.

I conclude that, in the hands of a capable designer, algorithmic modeling is a powerful tool which can fundamentally reframe the design process—but it is no substitute to that overall process. In the most direct sense, the designer is needed to define the problem formulation and interpret generated solutions. But in another sense, the designer should always be working above the algorithm—using human intuition to anticipate or react to the algorithm's flaws. This may involve fixing (or avoiding) bugs in the code, and keenly changing constraints as needed.

There is also the question of the substance of algorithmic designs. Taken on their own, algorithmic solutions tend to be narrowly focused. The subject of that foucs is optimized with precision, but does not necessarily encompass the broader design potential of an entire landscape. Users and audiences that are fresh to this approach may not be fully aware of this limitation, and may overestimate algorithmic design's comprehensiveness. It is easy to do so because of the highly interactive graphic interface and multi-functionality of algorithmic tools. Regarding that tendency, this project demonstrated to me that one should be careful not to conflate the complexity of the tool with the complexity of the design. Without the guiding hand of a designer, algorithmic solutions run the risk of being one-dimensional, and worse yet, boring. Only the designer can assign a unifying and interesting narrative to a landscape design. That role is not supplanted by the algorithmic modeling approach, but is strengthened by it.

References

Alexander, Christopher. “Systems Generating Systems.” *Architectural Design*. no. 38 (1968): 605 - 610.

Allen, Stan. “From Object to Field Architecture and Urbanism.” *Architectural Design*, no. 127 (1997): 24 - 31.

Ballard, B, S Wilson, H Udale-Clark, S Illman, T Scott, R Ashley, and R Kellagher. *The SuDS Manual*. London: Ciria, 2015.

Bassolino, Eduardo & Ambrosini, Luciano. “Parametric environmental climate adaptive design: the role of data design to control urban regeneration project of Borgo Antignano, Naples.” *Social and Behavioral Sciences*. No. 216 (2016): 948 - 959.

Belesky, Phillip. “Field Conditions.” Groundhog.la. 2016. Accessed December 2018. http://groundhog.la/techniques/field-conditions/

Belesky, Philip. “The Green Grasshopper.” *Journal of Digital Landscape Architecture*, no. 3 (May 30, 2018): 406 – 413.

Belesky, Philip. “Stacking Up.” *Landscapes Paysages*, Vol. 19, no 4 (November 25th, 2017): 46 - 49.

Cagan, Jonathan; Campbell, Matthew; Finger, Susan; Tomiyama, Tetsuo. “A Framework for Computational Design Synthesis: Model and Applications.” *Journal of Computing and Information Science in Engineering*. Vol. 5 (2005): 171 - 181.

Caliskan, Olgu. “Parametric Design in Urbanism: A Critical Reflection.” *Planning Practice and Research*. Vol. 32 No. 4 (2017): 417 - 443.

Cantrell, Bradley & Mekies, Adam. “Coding Landscape.” In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. New York, NY: Routledge (2018): 5 - 33.

Carpo, Mario. *The Digital Turn in Architecture 1992-2012*. West Sussex: John Wileyy & Sons Ltd., 2013.

Christoforetti, Elisabeth; Cohen, Will; Cohen, Yonatan; Rife, Stephen; Zhang, Jia. "Big Data for Small Places." In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. New York, NY: Routledge (2018): 5 - 33.

Chui, Ting Fong May; Liu, Xin; and Zan, Wenting. "Assessing Cost-Effectiveness of Specific LID Practice Designs in Response to Large Storm Events." *Journal of Hydrology.* Vol 533 (2016): 353 - 364.

City of Portland, *Portland Stormwater Management Manual, 2016*. Bureau of Environmental Services, Portland, Oregon (2016).

Cohn, David. "Evolution of Computer-Aided Design." *Digital Engineering.* Vol 247. December 2010. Accessed December 2018. https://www.digitalengineering247.com/article/evolution-of-computer-aided-design/

De Monchaux, Nicholas. *Local Code: 3,659 Proposals about Data, Design & the Nature of Cities*. New York, NY: Princeton Architectural Press (2016).

Ervin, Stephen. "Turing Landscapes: Computational and Algorithmic Design Approaches and Futures in Landscape Architecture." In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. New York, NY: Routledge (2018): 89 - 115.

Felson, Alex. "The Role of Models to Inform Green Infrastructure." Presentation at Simulating Natures Symposium, Philadelphia, PA, March 2015. Accessed January 2019. https://www.design.upenn.edu/landscape-architecture/events/simulating-natures

Fletcher, David. "The Parametric Park." In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. New York, NY: Routledge (2018): 71 - 78.

Forrester, Jay. *Urban Dynamics*. Cambridge, MA: MIT Press (1969).

Fricker, Pia; Girot, Christophe; and Munkel, George. "How to Teach 'New Tools' in Landscape Architecture in the Digital Overload," in Computation and Performance Proceedings of the 31st eCAADe Conference, ed. Rudi Stouffs and Sariyildiz. Delft University of Technology (2013): 545-553.

Girot, Christophe. "About Code." In *Codify: Parametric and Computational Design in Landscape Architecture*. Edited by Bradley Cantrell and Adam Mekies. New York, NY: Routledge (2018): 1 – 4.

Halpin, Margo and Johnson, Ben. "Latourette 30% Design Set," Portland: GreenWorks PC, 2019.

Jacobs, Jane. *Death and Life of Great American Cities*. New York: Vintage Books, 1963.

Kolarevic, Branko. *Architecture in the Digital Age: Design and Manufacturing*. New York: Spon Press (2003).

Owens, Larry. "Between Human and Machine: Feedback, Control, and Computing Before Cybernetics." *Technology and Culture*. Vol 44, No. 4 (2003): 843.

Schumacher, Patrik, and Leach, Neil. "Parametricism: A New Global Style for Architecture and Urban Design." *Architectural Design*. Vol 79, no. 4 (2009): 14 - 23.

Schumacher, Patrik. "Parametricism as Style—Parametricist Manifesto." Presented at 11th Architectural Biennale. Venice, Italy, 2008.

USDA National Resources Conservation Service, "Urban Hydrology for Small Watersheds: Technical Release 55," Washington D.C.: U.S. Department of Agriculture, 1986.

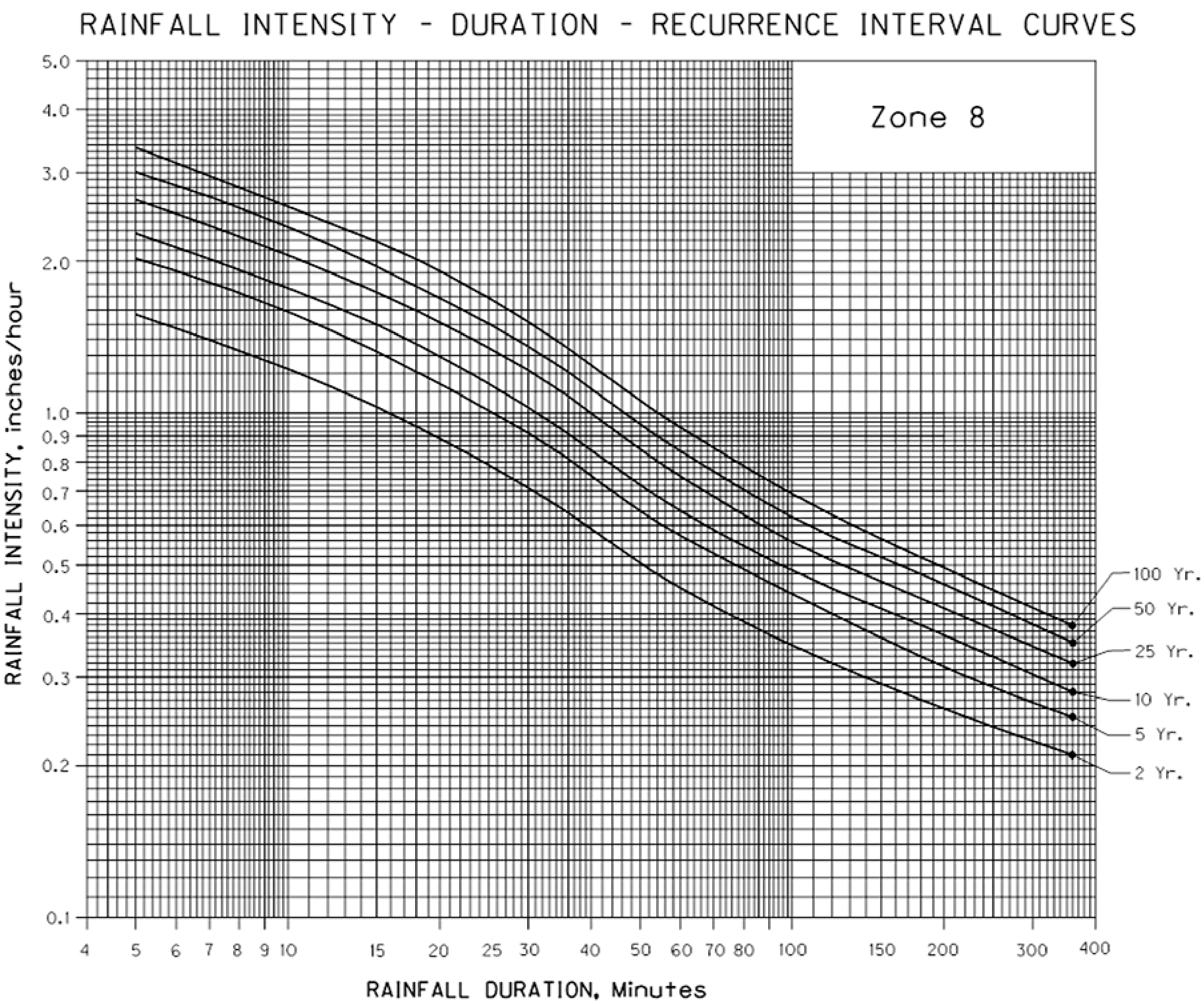
Walliss, Jillian, and Rahmann, Heike. *Landscape Architecture and Digital Technologies : Re-conceptualising Design and Making*. Abingdon, Oxon [UK] ; New York: Routledge, 2016.

Weaver, Warren. "Science and Complexity." *American Scientist*. no. 36 (1948): 536.

Appendix A: Design Storm

A.1//Rainfall Intensity Chart

Rainfall intensity is the rate of rainfall in inches per hour (in/hr) at it occurs at specific correlations between design storm frequency and the time of concentration. After having determined the time of concentration for a given drainage area, this standardized chart is used to determine the rainfall intensity, which subsequently factors into calculations of peak runoff rate and volume of runoff.



(Data and image sourced from ODOT. Accessed March 2019. https://www.oregon.gov/ODOT/GeoEnvironmental/Docs_Hydraulics_Manual/Hydraulics-07-A.pdf).

A.2//Design Infiltration Rates

The design and sizing of stormwater infrastructure components for infiltration largely depends on the infiltration rate of adjacent native soils. Ideally, infiltration rates are field tested as soil composition and compaction varies locally.

Generally, soil infiltration rates may be characterized according to hydrologic group and texture. The following table compiles that data. These are the preset values within the algorithmic hydologic calculations, and the data used in the execution of the Latourette Park site stormwater design.

Hydrologic soil group	Infiltration rate (cm/hr)	Infiltration rate (in/hr)	Soil texture
A	4.14	1.63	silty gravels, gravelly sands, sands
	2.03	0.8	sand, loamy sand, sandy loam
B	1.14	0.45	
	0.76	0.3	loam, silt loam
C	0.51	0.2	sandy clay, loam
D	0.15	0.06	clay loam, silty clay, sandy clay, clay

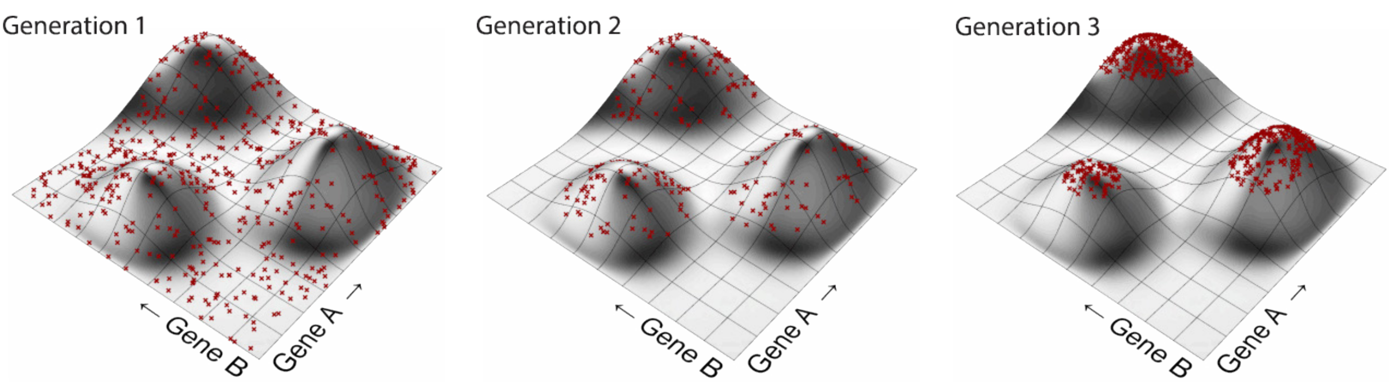
(Data adapted from USDA Natural Resources Conservation Service, “Urban Hydrology for Small Watersheds: TR-55,” June, 1986. Accessed March, 2019. https://hydrocad.net/pdf/TR-55%20Manual.pdf).

Appendix B: Galapagos

B.1//Galapagos Mechanics

Galapagos runs a progression of solution iterations, testing each solution and ostensibly improving subsequent solutions. The changes to each iteration occur within adjustable parameters that are given a particular range of variation. Iterations continue until the solutions can no longer be improved, potentially going through thousands of iterations before completing. Each iteration of Galapagos performs the following steps:

1. *Generation* — the first iteration generates a multitude of solutions, termed a population, from random permutations of adjustable parameters, termed genes. Subsequent generations are determined by the next steps.
2. *Fitness Evaluation* — each solution in the population is evaluated against predefined objective values. Fitness is the measure of how close the solution's value is to the target of the objective.
3. *Selection* — The individual solutions with the best fitness value are selected and sorted into pairs. Individuals with the poorest fitness value are not selected.
4. *Reproduction* — Each pair produces offspring that are a product of the parent generation's genes being recombined. The mechanisms of this process vary and are adjustable, but the basic result is a new solution with parameters that fall somewhere between the parents'.
5. *Replacement* — the initial population of solutions is eliminated and the new population starts the next iteration.



//Figure B1
Simplified demonstration of the evolutionary-solver process. The terrain represents the fitness value that results from an individual solution's unique combination of variables. This diagram only features two variables and three near-optimal solutions. (Dave Rutten, http://ieatbugsforbreakfast.wordpress.com).

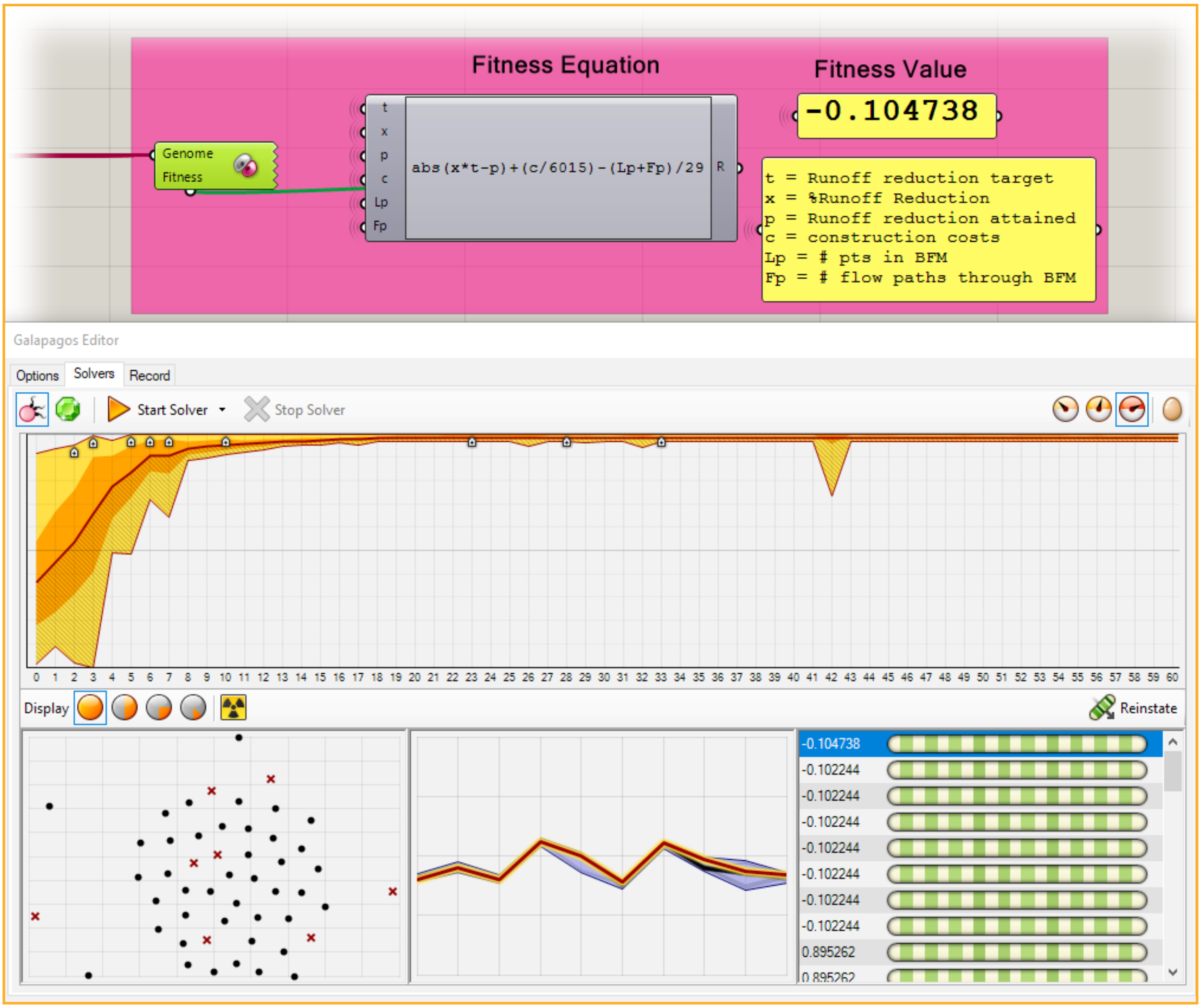
B.2//Solver Results

The Galapagos solver provides a user interface to control and monitor the solution process. As it progresses, the evaluation of each soltuon generation is displayed graphically. The various graphs portray the overall fitness value of the entire generation, the relative fitness values of individual solutions within the solution population, and the range of the solution variables.

The following are the results of the solver across each of the BFM solutions—two from Subcatchment 1 and two from Subcatchment 2. The image of the Galapagos readout is juxtaposed with the fitness equation—along with the fitness value of the optimized solution. A lower fitness value (near zero) indicates a higher fitness level.

i. Subcatchment 1, BFM 1A

ii. Subcatchment 1, BFM 1B



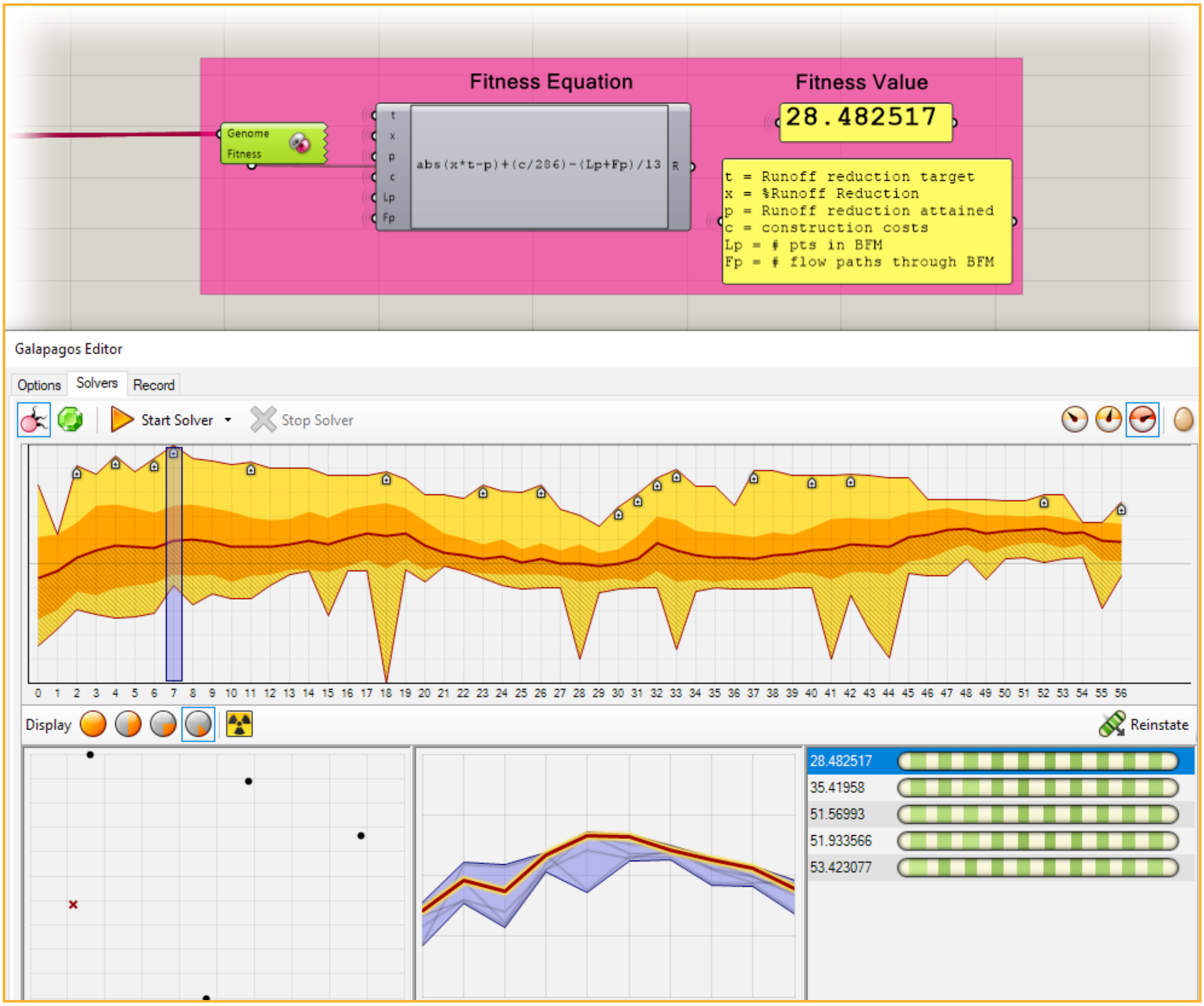
B2

B3

Appendix

iii. Subatchment 2, BFM 2A

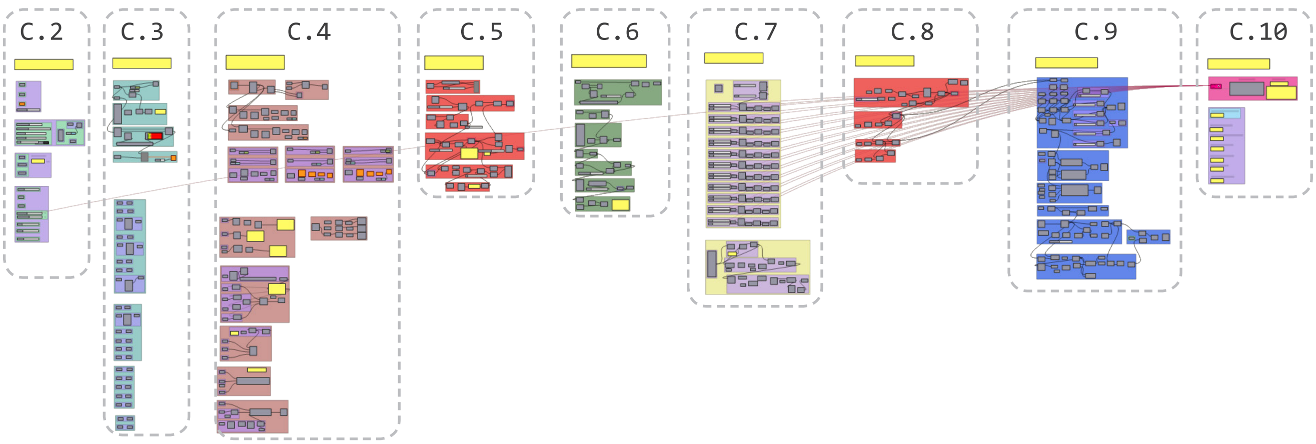
iv. Subcatchment 2, BFM 2B



Appendix C: Grasshopper Algorithm

The following section presents the completed Grasshopper algorithm for the generation of site-level stormwater solutions. For the sake of easing the develpment and use of this algorithm, it is compartmentalized into several conceptual steps and organized by spatial arrangement and color. The first diagram shows a coarse, full-view, and subsequent sections zoom in on individual steps to display inputs, components and connections on a more detailed level.

C.1//Completed workflow



C1

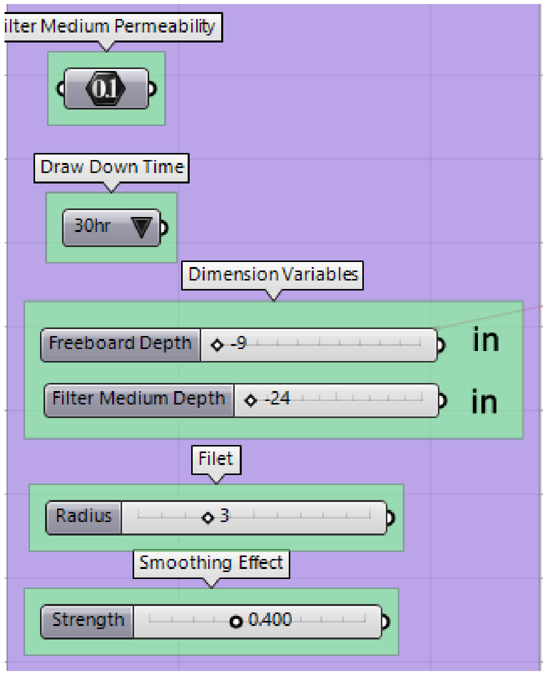
Workflow Index

C.2 User inputs and controls
C.3 Terrain analysis
C.4 Hydrological analysis
C.5 Solution domain placement
C.6 Solution domain constraints
C.7 BFM-solution generation
C.8 BFM 3D morphology
C.9 Solution performance analysis
C.10 Galapagos

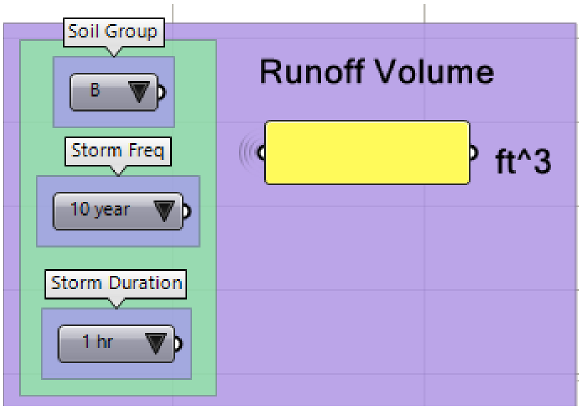
C.2//User controls and input parameters

Terrain Display

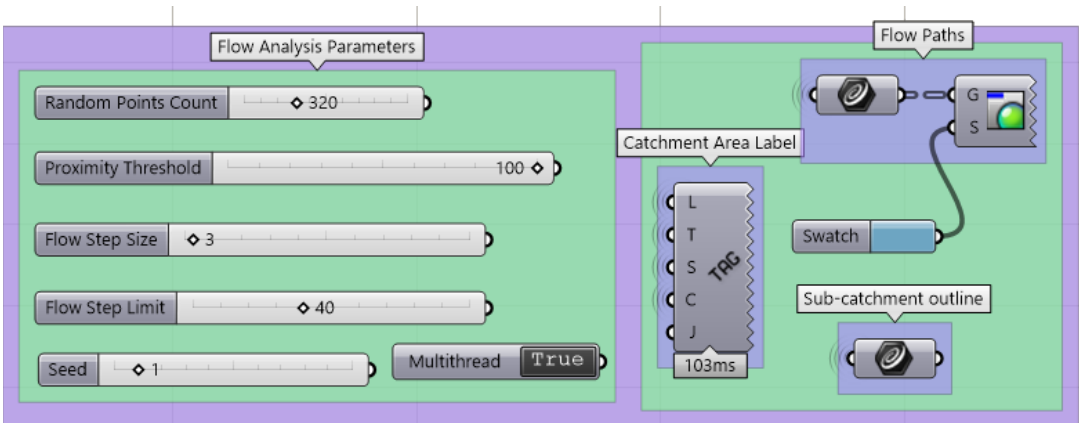
BFM Morphology Parameters



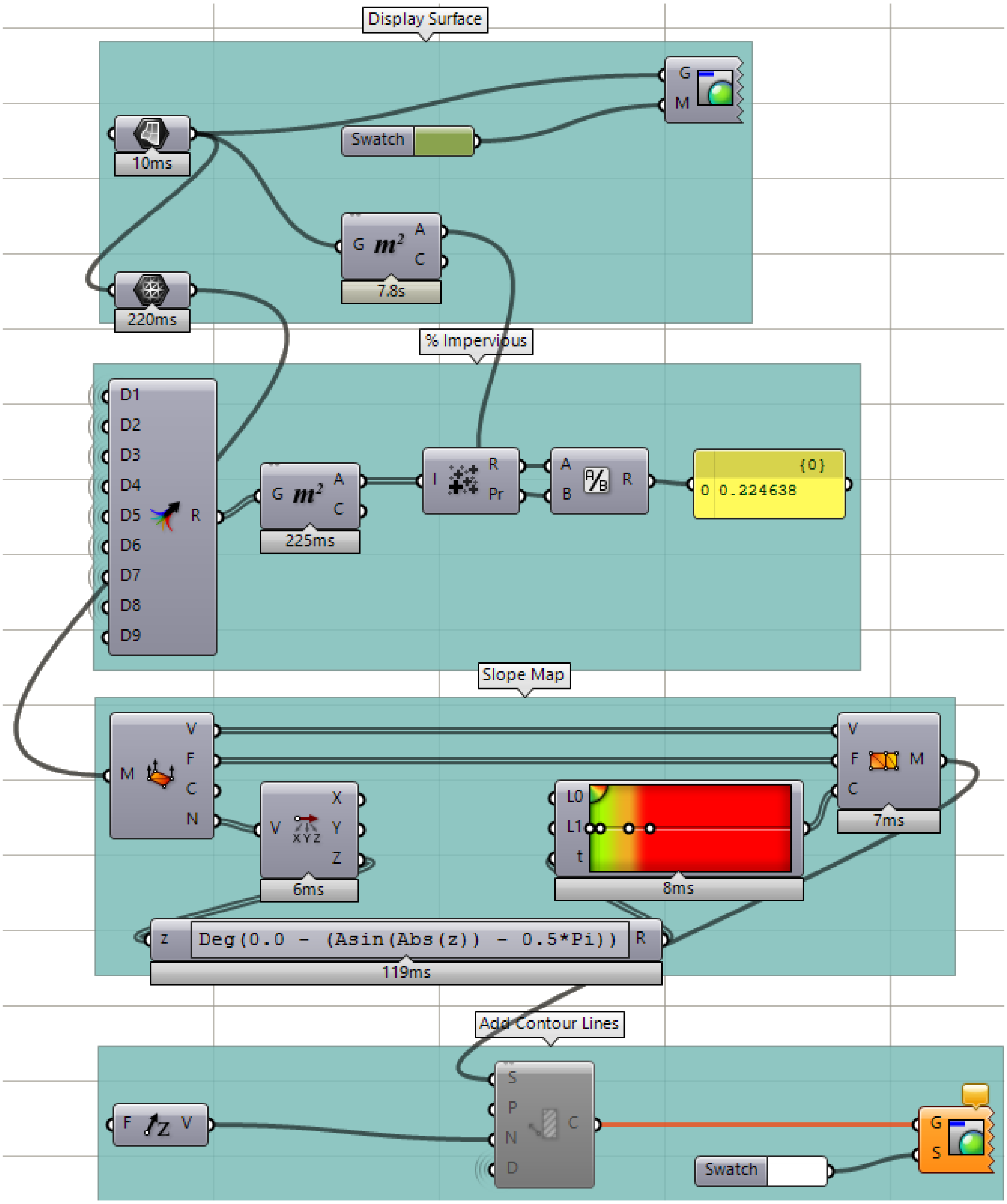
Design Storm Parameters



Hydrological Analysis Parameters

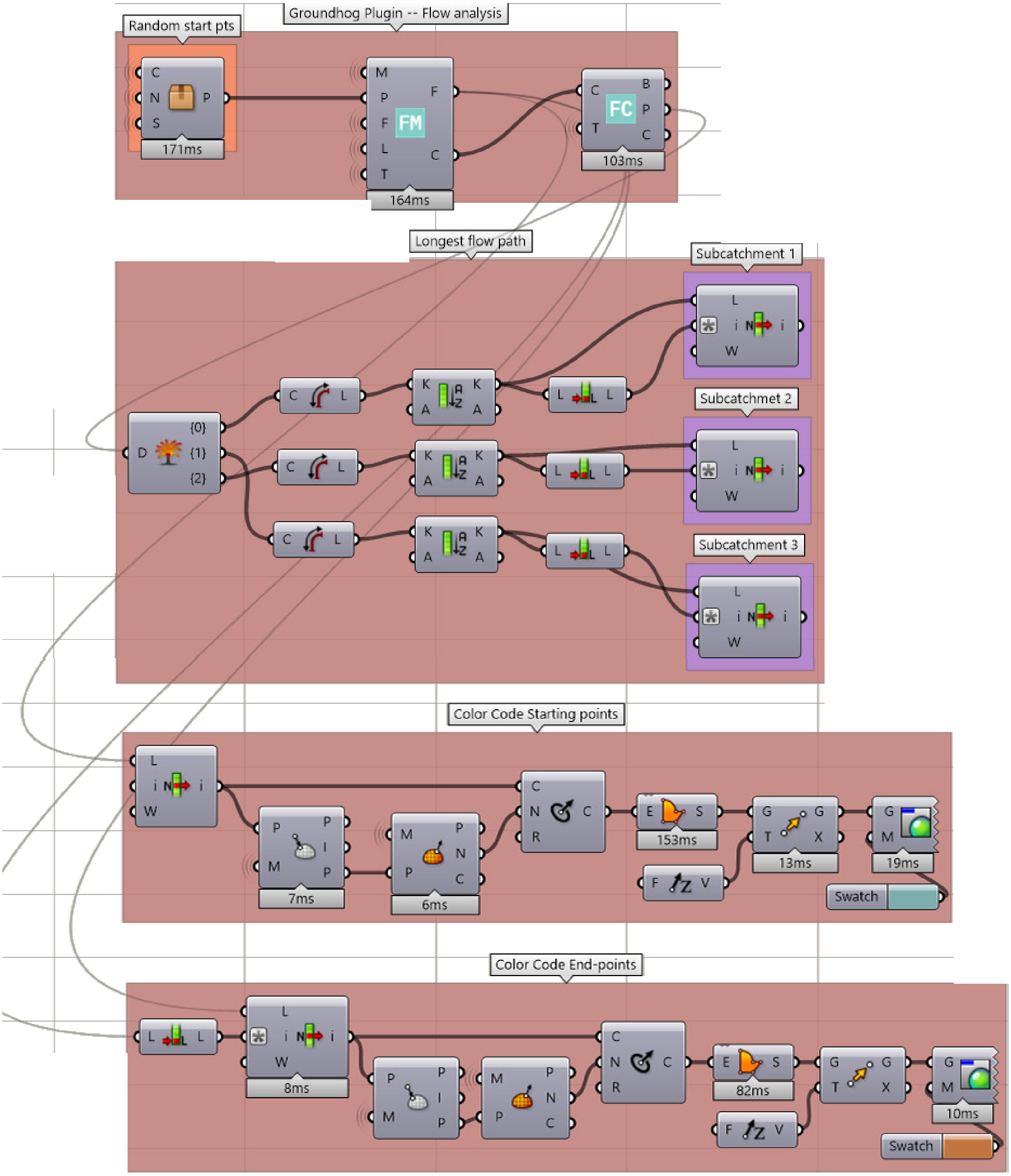


C.3//Terrain analysis

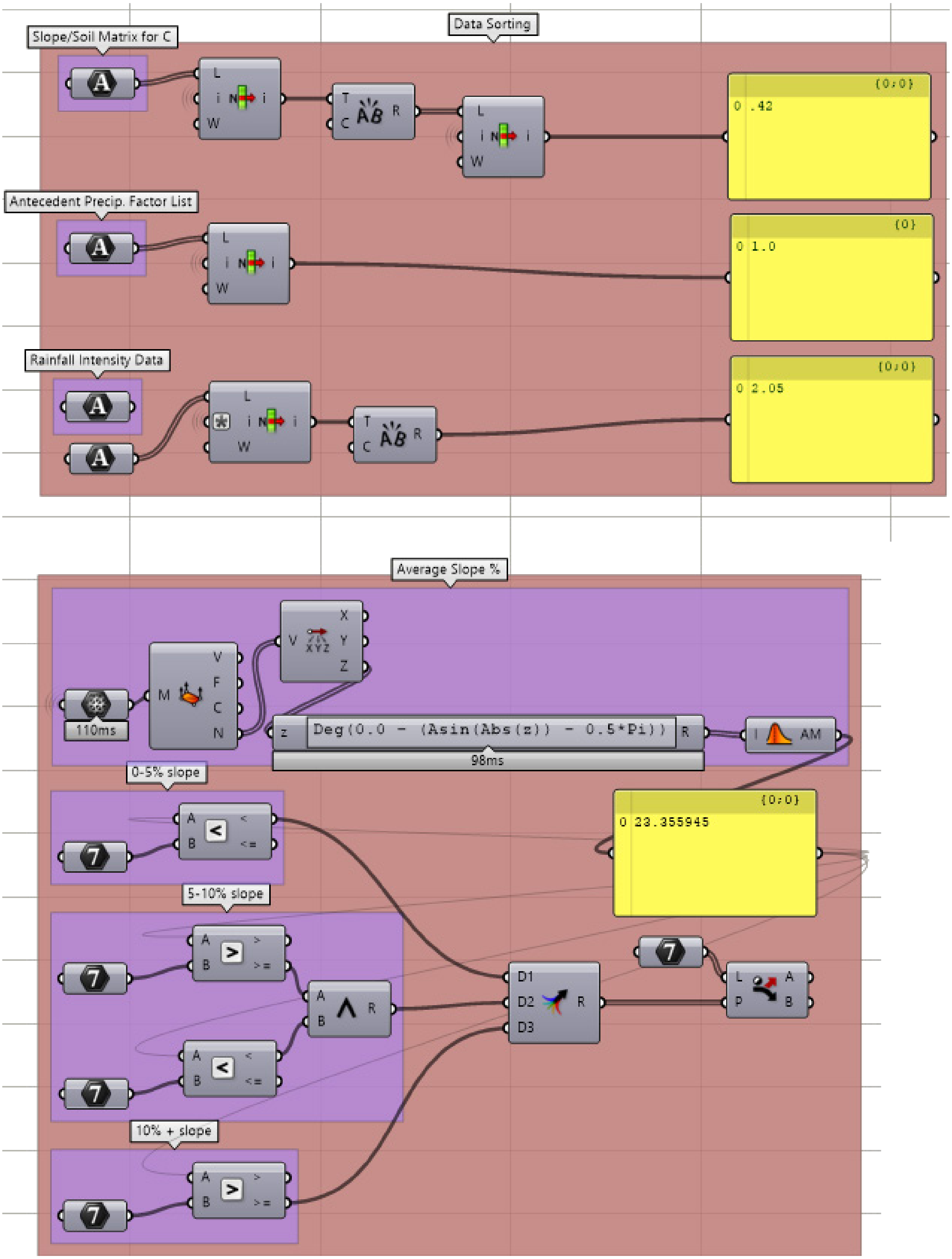


C.4//Hydrological analysis

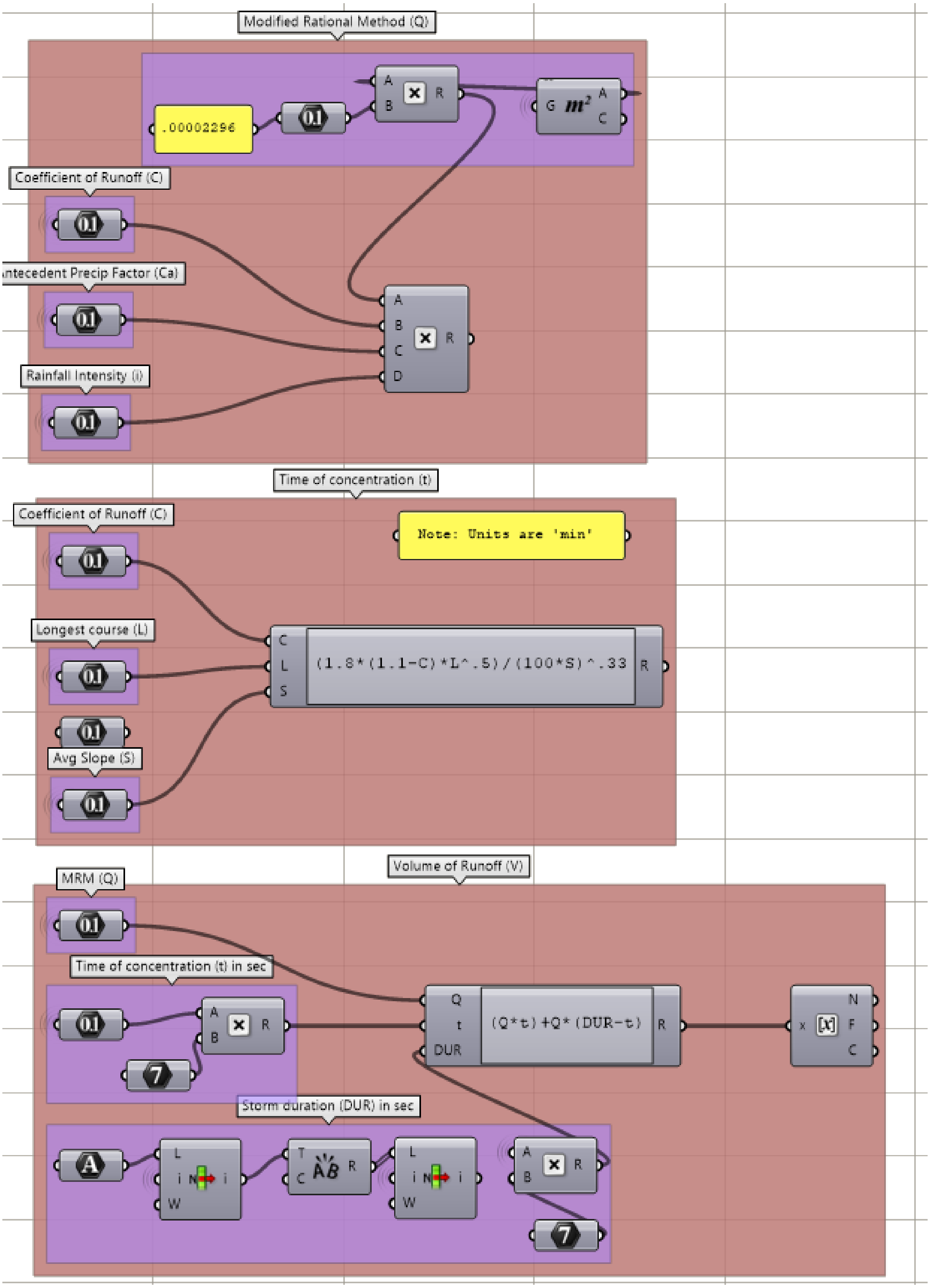
Surface flow simulation



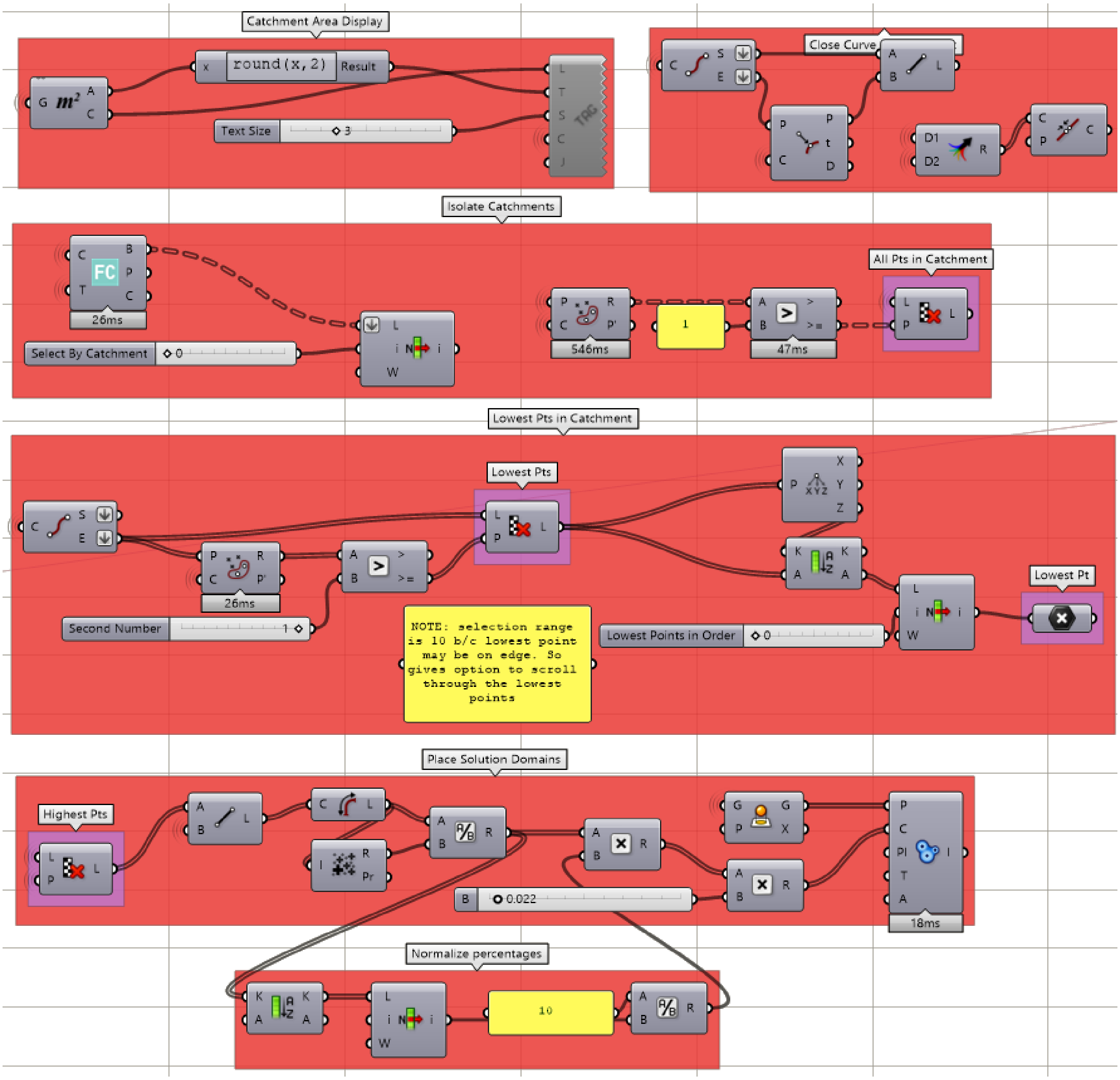
Subcatchment hydrological properties



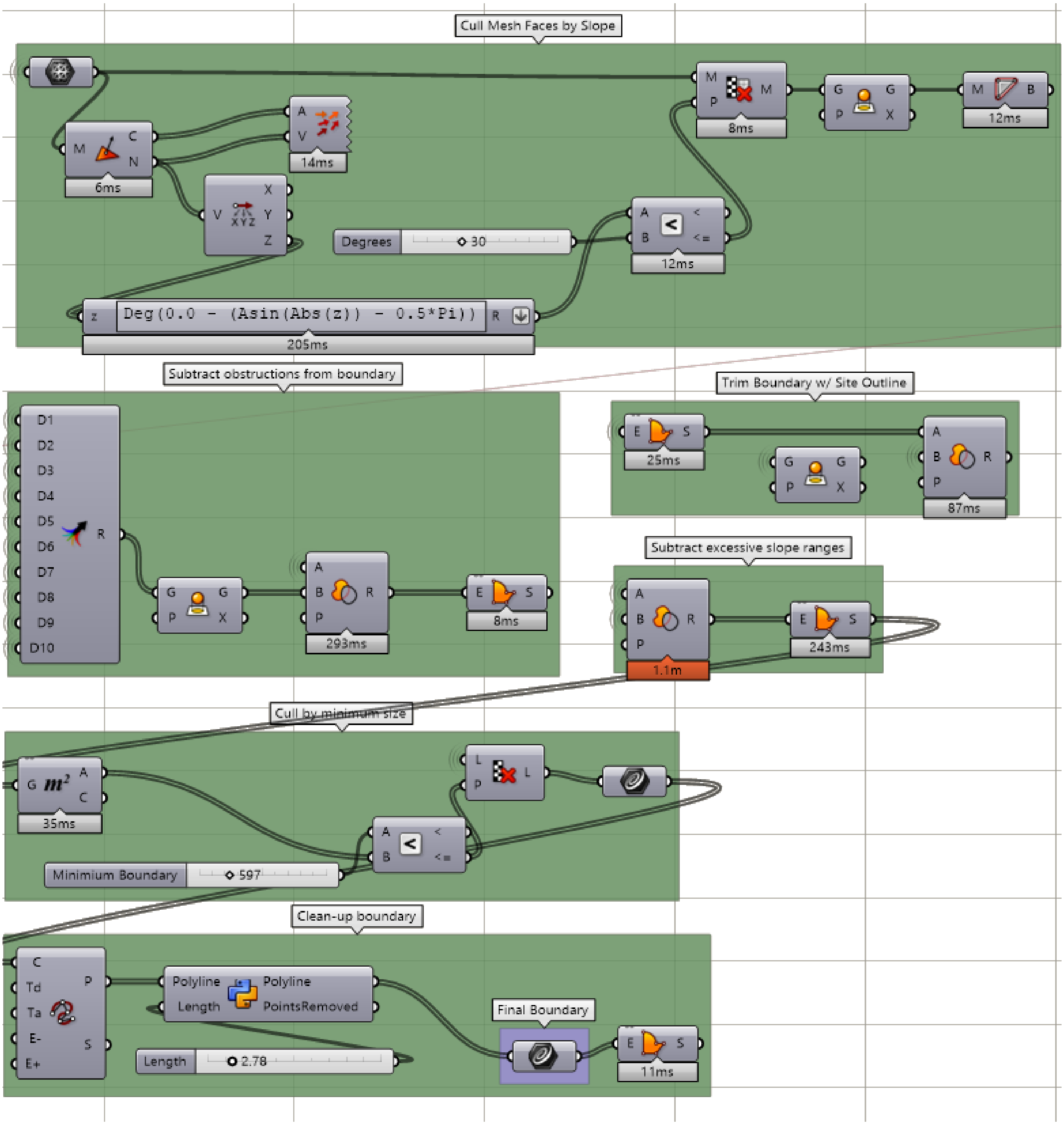
Runoff calculations



C.5//Solution domain placement

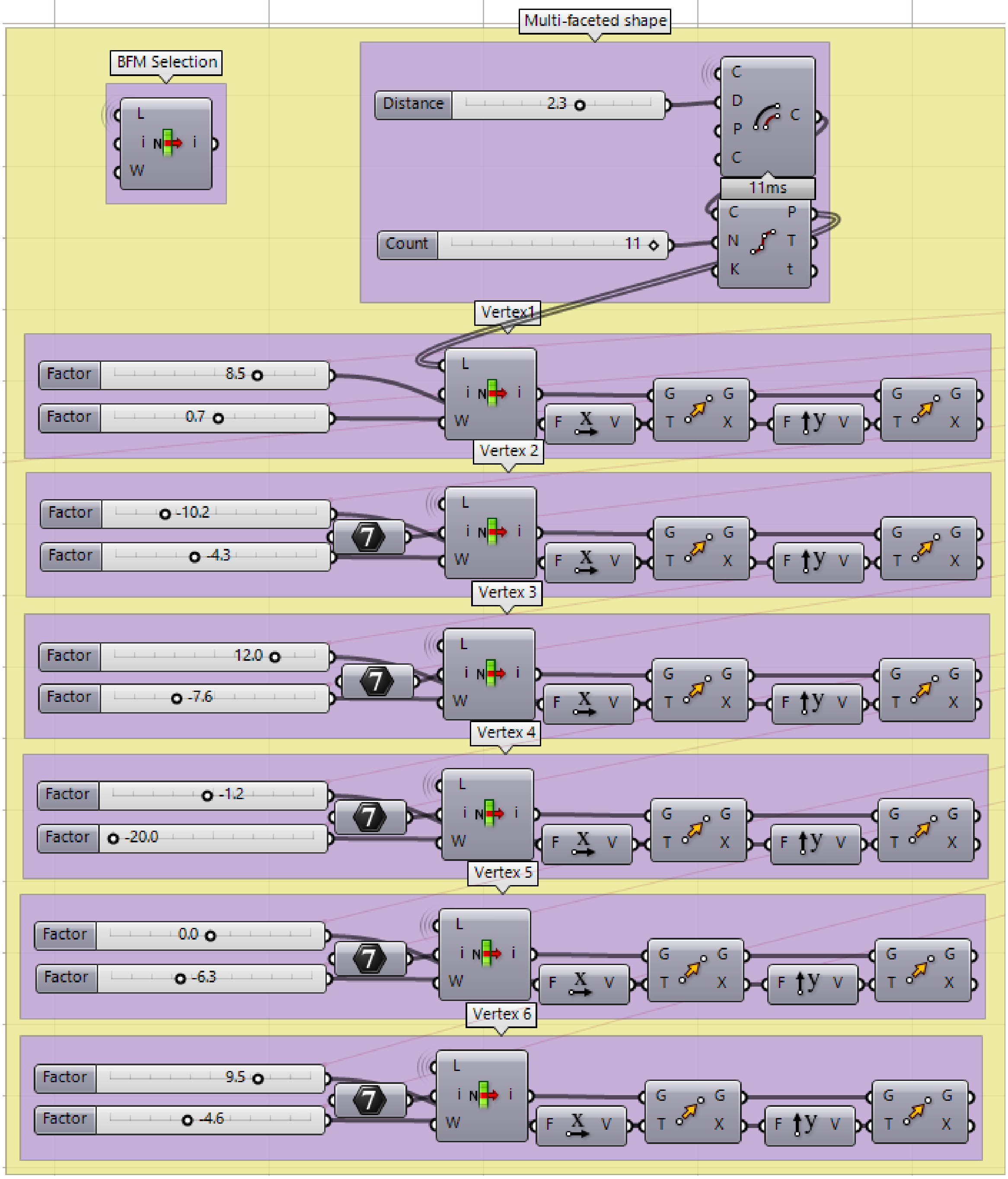


C.6//Solution domain constraints

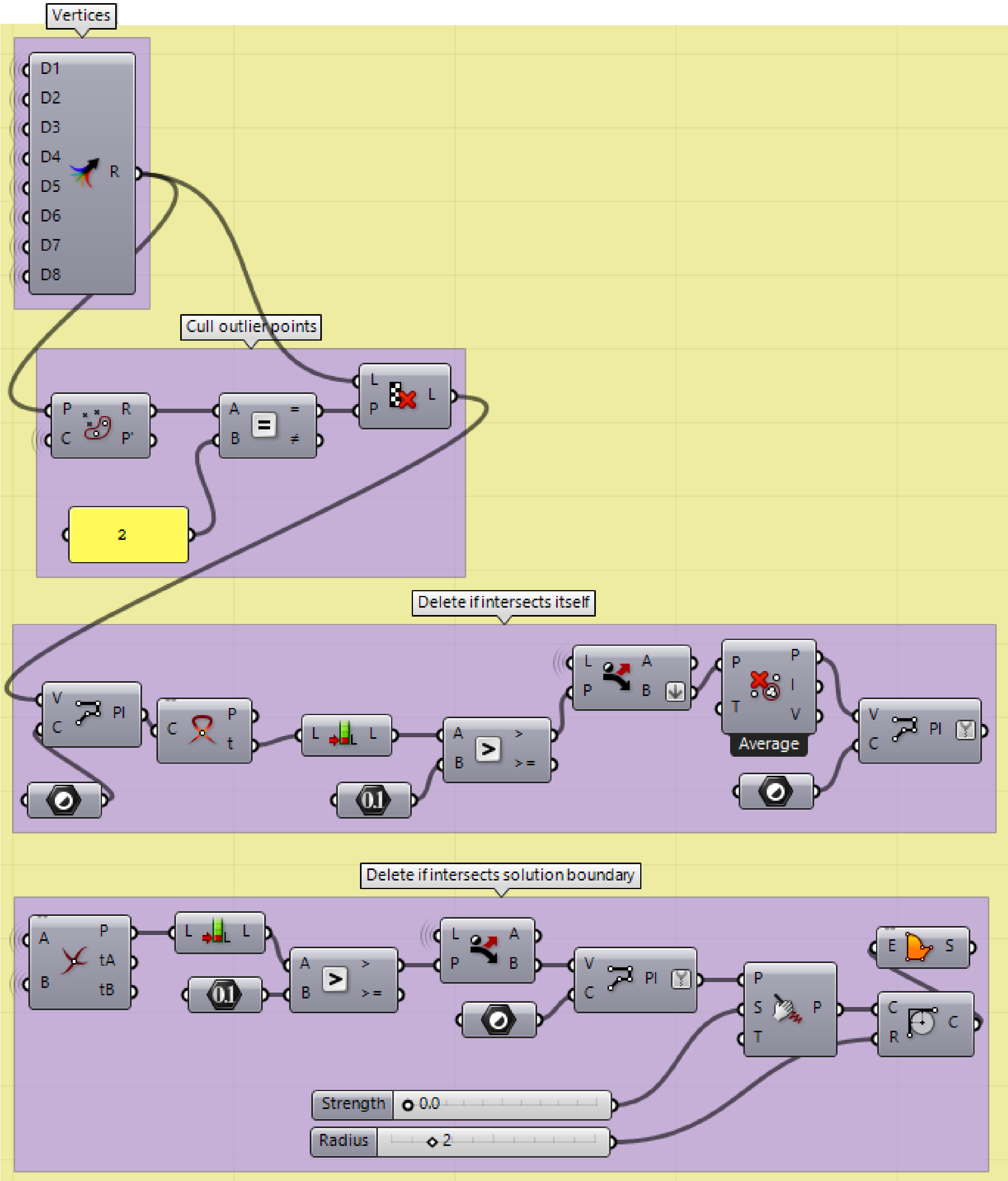


C.7//BFM-solution generation

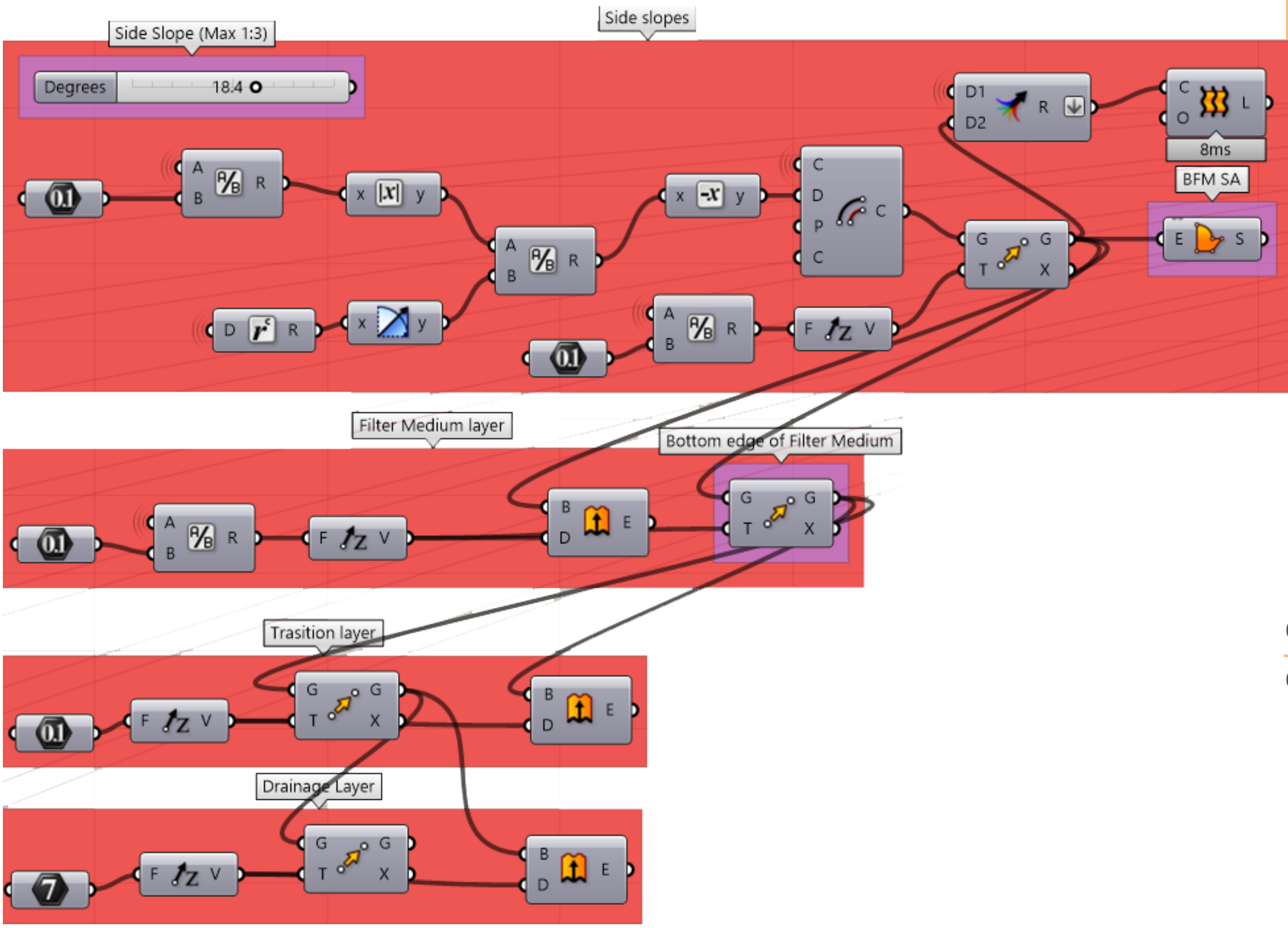
Vertex generation and movement



Null conditions

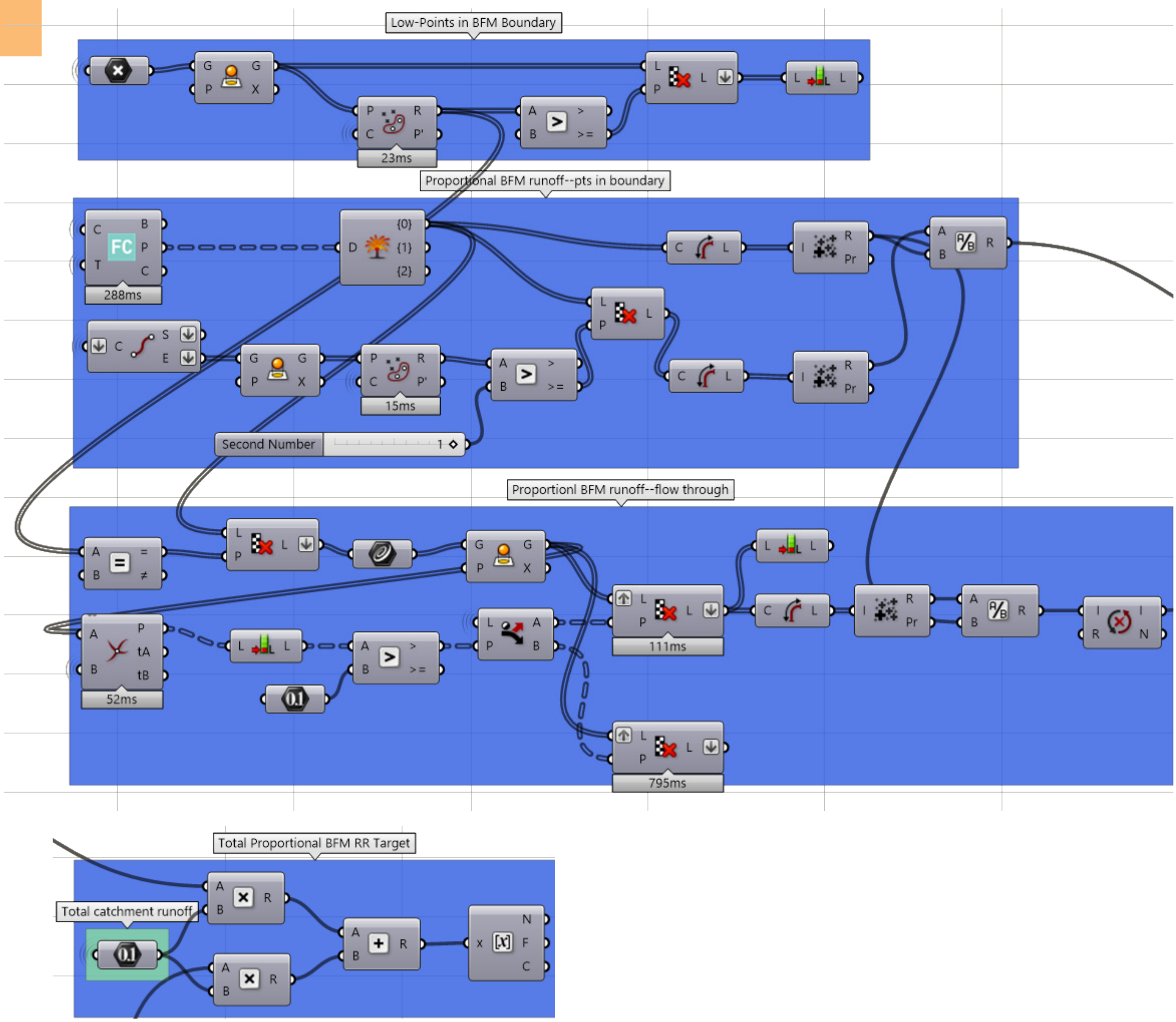


C.8//BFM 3D morphology

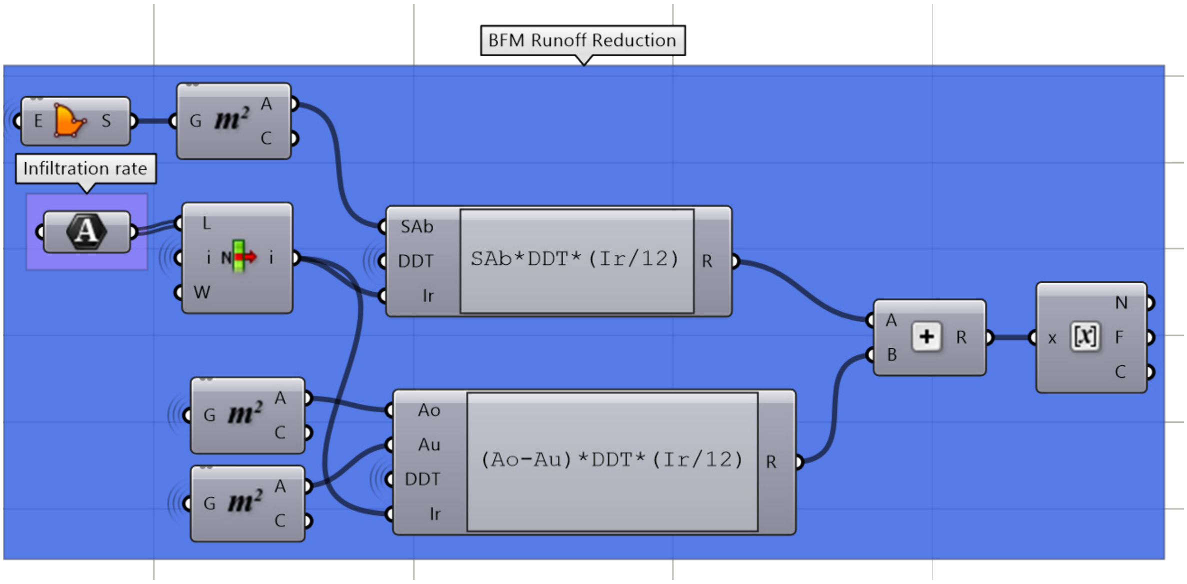


C.9//Solution performance analysis

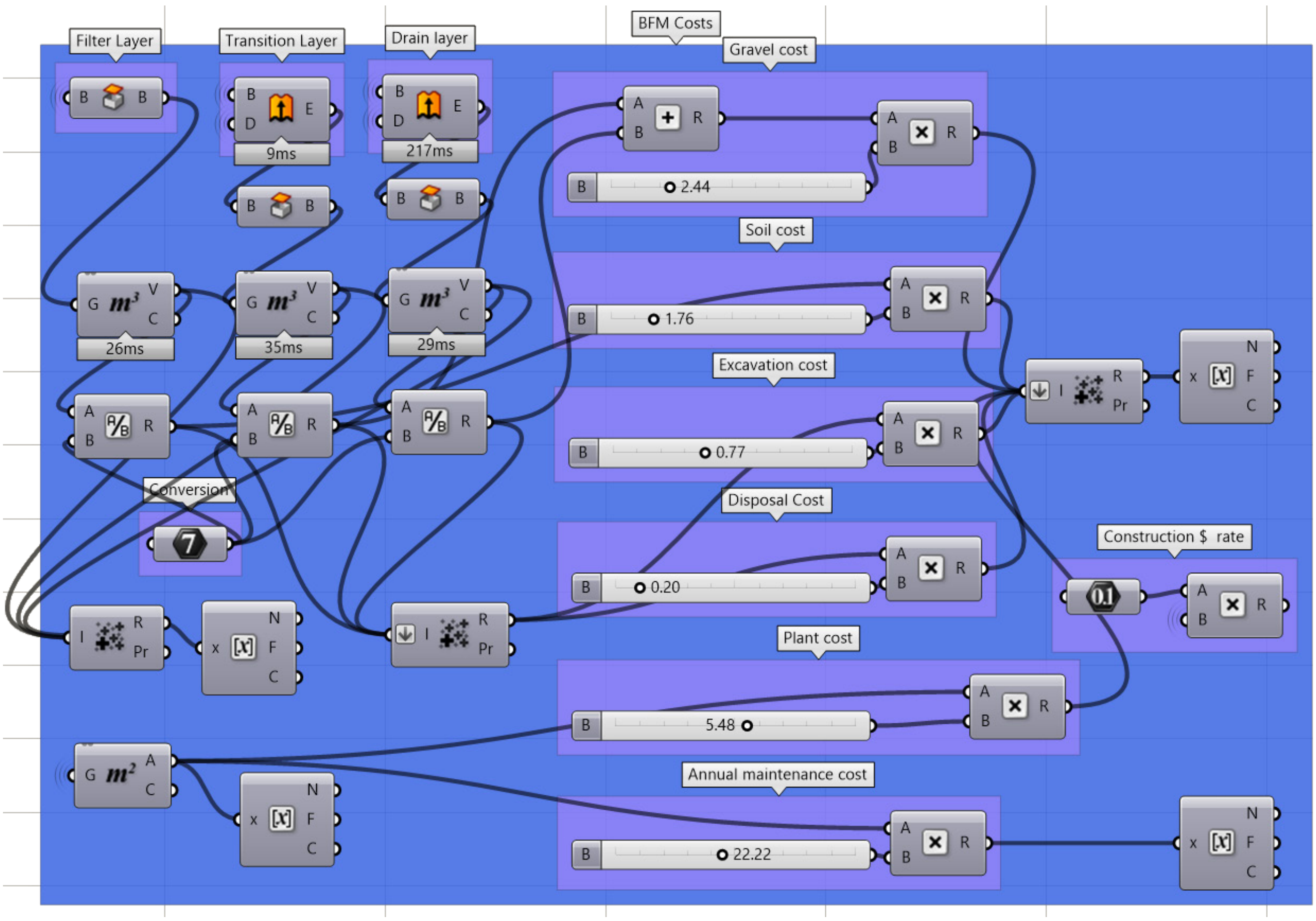
Calculation of Runoff Reduction target



Calculation of Runoff Reduction obtained

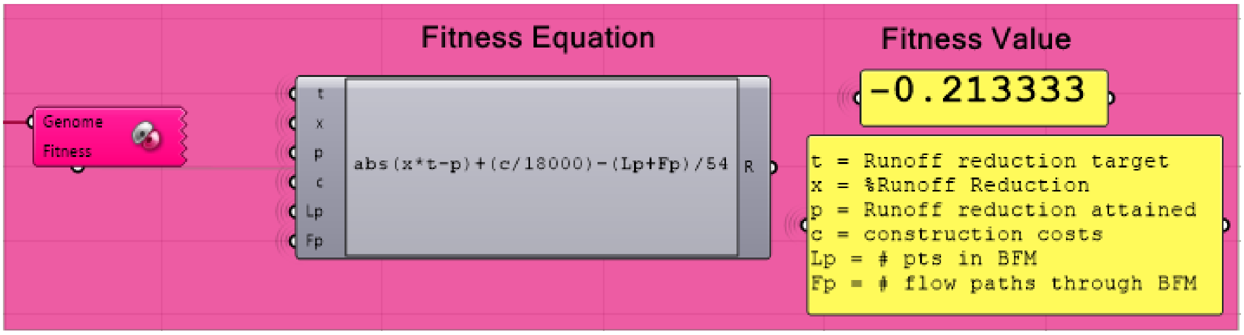


Cost calculations



C.10//Galapagos

Fitness equation



Performance results