

Statistical and physical characterization of rainfall-driven overland flow morphologies

Thèse N° 9269

Présentée le 26 avril 2019

à la Faculté de l'environnement naturel, architectural et construit
Laboratoire de technologie écologique
Programme doctoral en génie civil et environnement

pour l'obtention du grade de Docteur ès Sciences

par

Mohsen CHERAGHI

Acceptée sur proposition du jury

Prof. S. Takahama, président du jury
Prof. D. A. Barry, directeur de thèse
Prof. F. Schlunegger, rapporteur
Prof. P. Burlando, rapporteur
Prof. A. Rinaldo, rapporteur

2019

Drink wine and look at the moon
and think of all the civilizations
the moon has seen passing by.
—Omar Khayyam, The Rubaiyat of Omar Khayyam

Let the beauty of what you love be what you do
—Jalal ad-Din Muhammad Balkhi (Rumi)

To My Great Family...

Acknowledgements

I owe my great gratitude to the director of thesis, Prof. D. Andrew Barry, for his support. Andrew gave me the freedom to explore in my own way and at the same time, he guided me to the right directions of research. He always encouraged me to learn new skills and his precise reviews and critics substantially affected the quality of this dissertation. I am extremely grateful to Prof. Andrea Rinaldo who efficiently offered me the principal paths of the research. His ideas, comments and the strong interest in the topic are really appreciated. My sincere thanks go to Prof. Graham Sander for his support and constructive remarks, advice, and corrections. He greatly improved the quality of this work, especially the initial steps and defining the primary research questions. I also thank the jury members, Prof. Takahama Satoshi, Prof. Fritz Schlunegger and Prof. Paolo Burlando for their great comments and suggestions.

I would like to thank Prof. Paolo Perona, Dr. Seifeddine Jomaa and Dr. Andrea Cimattoribus who offered their generous help whenever I needed it. I am deeply thankful to Jacques Roland Golay, Antoine Wiedmer, Pierre-Alain Hildenbrand, Htet Kyi Wynn and Julian Barry who provided essential technical support for the execution of the experiments. I also appreciate the collaboration of SAGRAVE company, Lausanne, Switzerland.

I had a life-changing three-month trip to China during my PhD. I am appreciative to Prof. Pei Xin and Prof. Ling Li for hosting me at Hohai University and I thank Tingzhang Zhou and Prof. Guofen Hua for their support during my stay. I also appreciate the help of master students: Liu Lu, Yu Xiayang, Tang Honggen and Xu Haijun for their collaboration during the field measurements.

I would like to express my thanks to Marie Sudki, who apart from making the administrative parts so easy, was a great friend and I learned a lot from her for my life. I would like to extend my appreciation to my former and new colleagues and friends at ECOL who provided an excellent and delightful atmosphere for me during my doctoral study: Amir Razmi with whom I shared many beautiful memories, Elisa

Acknowledgements

Bruque Pozas who was my first lovely officemate and I really miss her, Abolfazl Irani for his kindness and the nice memories we made, Rafael Reiss for his humor, friendliness and nice coffees(!), Jiang Qihao for his cute behaviors, Frédéric Soullignac, Mahmood Ziabari, Mehrshad Foroughan, Benjamin Graf, Alexandre Serex and Karim Luy for the great times we shared at ECOL. During these years, a hidden source of energy was to chat with amazing people during the coffee breaks: Florian Breider, Virginie Bachmann, Camille Wolf, Margot Olive, Simon Meister, David Scheidweiler, Tania Gonin and Paraskevi Pramateftaki.

I am so grateful to many great friends whom I have been fortunate to meet and share enjoyable moments; particularly, Amir Razmi and Rozita Moradi, Abolfazl Irani and Mohaddese Jandaghian, Nima Aliakbari and Sahar Aliakbari, Milad Tatari, Dash Majid agha Balaei, Mojtaba Abdi, Naser Nasrollahzadeh, Hamed Izadi, Mahdad Kamali, Luvé Roman, Ali Khatir, Kamal Shahim, Shahb Eghbali, Masoud Alipour, Michèle Heeb, Homaira kamali, Amin Nyayifar, Mohammad Moosavi, Soheil Hassanzadeh, Sina Shamsoddin, Mohammadreza Saeedmanesh, Mehran Khaghani, Soroush Shafizadeh, Kamran Kalbasi, Saeed Omid, Mehdi Jafari, Amir Yazdani, Ali Motamed, Amin Shoaie, Mehdi Khorramshahi and Gérard Guffon.

Leïla est ma plus belle découverte à Lausanne et ma jolie Biche. Thank you Leïla for all the beautiful days, weekends and travels we made. Your presence made the days of my PhD very easy.

Last but not least, I am not able to thank my family by some words for all their support, care and love. I dedicate this thesis to my parents, my lovely sisters, Soudabeh and Fatemeh, and my niece Mehrsa.

Lausanne, 2 July 2018

Mohsen Cheraghi

Abstract

Environmental transport processes are highly coupled with the shape of landscapes. Modern catchment analysis with high-resolution data and huge computational powers demand more detailed within-cell metrics for surface evolution modeling. This dissertation involves experiments and numerical simulations of rainfall-driven sediment transport at laboratory scales of which areas were less than the common computational cell sizes at catchment scales. The main objective was stochastic and physical study of unchanneled overland flow morphologies.

In the first step, a detailed laboratory study was conducted to highlight the effects of morphological changes on hysteretic sediment transport under a time-varying rainfall. A rainfall pattern composed of seven sequential stepwise-varying rainfall intensities was applied to a 5-m \times 2-m soil erosion flume. Clockwise hysteresis loops in the sediment concentration versus discharge curves were measured for the total eroded soil and the finer particle sizes. However, for larger particle sizes, hysteresis effects decreased and suspended concentrations tended to vary linearly with discharge. The Hairsine and Rose (HR) soil erosion model agreed well with the experimental data for the total eroded soil and for the finer particle size classes. For the larger sediment size classes, the model provided reasonable qualitative agreement with the measurements although the fit was poor for the largest size class. Overall, it is found that hysteresis varies amongst particle sizes and that the HR model's results are consistent with hysteretic behavior of different sediment size classes.

After demonstrating the role of morphological changes (generation of a shield layer on the topsoil) on erosion patterns, overland flow morphologies were statistically characterized. To do so, the catchment-scale network analyses were applied on the micro-roughnesses of unchanneled surfaces at the laboratory scale (2-m \times 1-m flume). The scaling relation between the drainage area and stream length (Hack's law), along with exceedance probabilities of drainage area, discharge, and upstream flow network length, are well known for catchment-scale channelized fluvial regions.

Abstract

This work reported catchment-scale power laws on an eroding unconsolidated sediment for which no channeling occurred. Laser scanning was used to capture the morphological evolution of the sediment. Based on the surface scans and precipitation distribution, overland flow was estimated with the D8 algorithm, which outputs a flow network that was analyzed statistically. The above-mentioned scaling and exceedance probability relationships for this overland flow network were the same as those found for large-scale catchments and for laboratory experiments with observable channels. In addition, the scaling laws were temporally invariant, even though the network dynamically changed over the course of the experiment.

The statistical similarity of a small-scale unchanneled morphology to catchments motivated us to test the applicability of a physically-based catchment scale landscape evolution model (LEM) at laboratory scale and in absence of rills. We modeled the overland flow as a network that preserves the water flux for each cell in the discretized domain. This network represented the surface flow and determined the evolution direction. The calibrated model predicted a smoother surface morphology (and less detailed overland flow network) than that which was measured, since the experimental soil contained small pebbles that emerged during erosion, and were delineated by the laser scans. To investigate the quality of the prediction, we used a low-pass filter to remove the small-scale variability of the surface morphology. Spectral analysis confirmed that the model predictions capture the main characteristics of the measured morphology. However, the model could not reproduce the experimental scaling relation as the micro-roughness of the surface was not produced by the model. In order to modify the LEM, it was solved as a stochastic partial differential equation. The results showed that an extra term for roughness was necessary to simulate more details of the morphology. Furthermore, a new deterministic approach (diffusion coefficient as a step function of curvature) was proposed and tested to improve the model predictions in the statistical sense.

This work showed the hysteresis loops for splash dominated erosion and the different hysteretic behavior of particle sizes. Furthermore, the advantageous statistical metrics (scaling laws) were found for unchanneled morphology evolutions and the applicability of an LEM was shown in absence of rills. Besides the scientific contributions, useful modeling, optimization and data analysis tools (C++ and Python codes) were provided for future geomorphological studies.

Key words: Overland flow, Sediment transport, Landscape evolution model, Hairsine

and Rose (HR) erosion model, Network evolution, Statistical analysis, Numerical simulation, Spectral analysis, Optimization, C++, Python.



فرایندهای انتقال در محیط زیست به شدت به مشخصه‌های هندسی محیط وابسته است. تجزیه و تحلیل‌های زیست محیطی مدرن با داده‌های با وضوح بالا و قدرت محاسباتی چشم‌گیر، اندازه‌گیری و پیش‌بینی دقیق‌تر مشخصات سطحی زمین را می‌طلبد. این پایان‌نامه شامل تحلیل فیزیکی و آماری تغییرات سطح ناشی از بارندگی است به طوری که مساحت مورد نظر در آزمایشات و شبیه سازی‌های عددی کمتر از یک سلول محاسباتی در مطالعه‌های میدانی می‌باشد.

در مرحله اول، یک مطالعه دقیق به منظور اثرات پسماند^۱ در انتقال رسوب تحت بارندگی متغیر با زمان انجام شد. یک الگوی بارندگی شامل هفت مرحله متوالی با شدت بارش متغیر روی سطح اعمال شده و غلظت ذرات معلق با اندازه‌های مختلف اندازه‌گیری شد. پس از بررسی نتایج، حلقه‌های هیستریزیس ساعتگرد برای غلظت کل رسوبات و همچنین غلظت ذرات ریز پدیدار آمد. در مقابل، با افزایش اندازه ذرات، اثرات هیستریزیس کاهش و رابطه غلظت ذرات معلق با دبی خروجی آب خطی به نظر رسید. مدل فرسایش هیرسین-رز^۲ با اندازه‌گیری‌های تجربی بهینه شده و نتایج مقایسه شد. مدل بهینه نتایج مشابهی را برای غلظت کل و غلظت ذرات ریز نشان داد. در حالی که نتایج برای ذرات بزرگتر با داده‌های آزمایشگاهی متفاوت بود، روند تغییرات به درستی شبیه‌سازی شد. به وسیله این مدل، غلظت ذرات مختلف در لایه محافظ اندازه‌گیری شد. نتایج مدل نشان داد که ته‌نشینی ذرات بزرگتر روی سطح موجب کمتر شدن منبع ذرات کوچک‌تر و در نتیجه حلقه‌های هیسترتیک^۳ می‌شود. این مطالعه نشان داد که تغییرات ناچیز روی سطح موجب اثرات چشم‌گیر در روند فرسایش می‌شود.

پس از نشان دادن اهمیت تغییرات جزئی سطح، در مرحله بعد مطالعه آماری هندسه سطح انجام شد. برای این منظور از تحلیل‌های شبکه رودخانه‌ها (قانون هک^۴؛ توزیع احتمالی مساحت زهکشی و توزیع احتمالی طول بالادست) استفاده شد. با وجود عدم تشکیل

¹ Hysteresis

² Hairsine-Rose

³ Hysteretic

⁴ Hack's law

میکروکانال روی سطح، پروفایل آن با لیزر و با دقت بالا اندازه‌گیری شده، سپس با استفاده از الگوریتم دی-۸^۵ و پراکندگی بارندگی، شبکه جریان روی سطح بدست آمد. این شبکه به عنوان ابزاری برای تحلیل آماری مورد استفاده قرار گرفت. نتایج این مطالعه حاکی از وجود قانون هک و هم چنین توزیع‌های احتمالی مشابه در جزئیات زبری سطح بود. همچنین، حتی با تغییرات دینامیک شبکه جریان، این معیارهای آماری در طول زمان پایدار ماند. بنابراین، می‌توان مشخصه‌های آماری شبکه‌های رودخانه‌ای (در مقیاس ده‌ها تا هزاران کیلومتر) را روی سطح تحت بارندگی (در مقیاس یک متر) مشاهده نمود. این ویژگی، یک معیار بسیار مفید برای تغییرات سطحی زمین در مقیاس‌های کوچک معرفی می‌کند که برای توسعه مدل‌های فرسایش بسیار کاربردی خواهد بود.

شباهت آماری زبری سطح و شبکه‌های رودخانه‌ای ما را بر آن داشت که از مدل‌های فیزیکی توسعه شبکه‌های رودخانه‌ای به منظور شبیه‌سازی سطوح کوچک تحت بارندگی و در غیاب کانال استفاده کنیم. بدین منظور مدل متداول جابجایی-پخش با سطوح اندازه‌گیری شده (در مدت ۸ ساعت بارندگی) بهینه شده، تغییرات سطح در ۸ ساعت بعدی توسط مدل پیش‌بینی شده و با نتایج واقعی مقایسه شد. این نتایج حاکی از آن بود که مدل استفاده شده قابلیت پیش‌بینی تغییرات کلی سطح را دارد. هر چند که تحلیل‌های فوریه نشان داد که مدل در شبیه‌سازی جزئیات سطح (فرکانس‌های بالا و معادل حدود ۱۰ میلی متر) دارای محدودیت است. به عبارتی، مدل سطح فیلتر شده را شبیه‌سازی نمود.

به منظور ارتقای مدل فیزیکی برای زبری‌های فرکانس بالا، به جای استفاده از ضرایب همگن در معادله دیفرانسیل، در شش طراحی‌های مختلف از ضرایب تصادفی استفاده شده و یک ترم زبری تصادفی نیز به معادله اضافه شد. علاوه بر این، یک رویکرد تحلیلی جدید (در نظر گرفتن ضریب پخش به عنوان تابعی از انحنای موضعی) پیشنهاد شده و با داده‌های تجربی امتحان شد. در مجموع، از میان ۷ راهکار مختلف، دو روش برای بهبود نسبی مدل توصیه شد.

به طور خلاصه، این مطالعات نشان داد که علاوه بر جریان رودخانه‌ها، حلقه‌های هیستریزیس برای فرسایش ناشی از بارندگی نیز وجود دارند و شدت اثر هیستریزیک تابعی از اندازه ذرات است. علاوه بر این، در صورت استفاده از داده‌هایی با دقت بالا، قوانین آماری و فیزیکی شبکه‌های رودخانه را می‌توان برای سطوح تحت بارندگی و در غیاب میکروکانال‌ها مورد استفاده قرار داد.

علاوه بر نتایج علمی فوق، کدهای شبیه‌سازی، بهینه‌سازی و تجزیه و تحلیل داده‌ها در محیط‌های سی-پلاس-پلاس^۶ و پایتون^۷ نوشته شده و به صورت منبع آزاد برای پژوهش‌های آینده ارائه شده است.

کلید واژه‌ها: شبیه‌سازی عددی، آنالیز آماری، مدل توسعه شبکه رودخانه، جریان آب سطحی، فرسایش ناشی از باران، مدل هیرسین-رز، تحلیل فوریه، سی-پلاس-پلاس و پایتون.

^۵ D8 algorithm

^۶ C++

^۷ Python

Résumé

Les processus de transport dans l'environnement sont fortement couplés à la géométrie du paysage. Les études récentes sur les bassins versants réalisées à partir de données hautes résolutions et mettant en oeuvre d'énormes puissances de calcul exigent des métriques plus précises à l'intérieur des mailles de calcul pour modéliser l'évolution des surfaces. Cette thèse associe des expérimentations et des simulations numériques du transport de sédiments par les précipitations à l'échelle du laboratoire dont la superficie était inférieure à la taille des mailles de calcul dans les modélisations de bassins versants complets. L'objectif principal était l'étude stochastique et physique de la morphologie d'écoulements non canalisés.

Premièrement, une étude détaillée en laboratoire a été réalisée pour souligner les effets des changements morphologiques sur le transport de sédiments hystérétique dans des conditions de précipitations changeante. Un régime de pluie composé d'une séquence de sept intensités différentes a été appliquée à un simulateur de pluie de 5 m par 2 m. Des boucles d'hystérésis de concentration en sédiments tournant dans le sens horaire par rapport à des courbes de débit ont été mesurées pour l'érosion totale du sol et pour les particules les plus petites. Cependant, pour des particules plus grandes, l'hystérésis diminue et la concentration tend à varier de façon presque linéaire avec le débit. Le modèle d'érosion Hairsine-Rose (HR) s'accorde bien avec les données expérimentales d'érosion totale pour les classes de particules les plus fines. Pour les classes de particules plus grandes, le modèle a donné des résultats satisfaisants par rapport aux mesures bien que l'adéquation soit pauvre pour les classes de taille les plus larges. Dans l'ensemble, il a été démontré que l'hystérésis varie parmi les tailles de particule et que les prédictions du modèle HR sont cohérentes avec le comportement hystérétique de différentes classes de taille de sédiments.

Après avoir démontré le rôle des changements morphologiques (création d'une

couche de protection à la surface du sol) de patterns d'érosion, les morphologies de ruissellement ont été caractérisées de manière statistique. Pour cela, les études réalisées sur le réseau à l'échelle du bassin versant ont été appliquées à la micro-rugosité des surface non-canalises à l'échelle du laboratoire (simulateur de pluie de 2 m par 1 m). Les relations d'échelles entre l'aire de drainage et la longueur de flux (loi de Hack), avec aussi les probabilités de dépassement de l'aire de drainage, du débit et de la longueur de l'écoulement dans le réseau à l'amont, est bien connu pour les portions de fleuve canalises. Ce travail couvre l'érosion de sédiments non consolidés dans un environnement non-canalise. Le balayage laser a été utilisé pour mesurer l'évolution morphologique des sédiments. A partir des scans de la surface et de la distribution des précipitations, le ruissellement a été estimé avec l'algorithme D8 qui donne en sortie un réseau d'écoulement qui a été analysé de manière statistique. La mise à l'échelle et les relations de probabilité de dépassement susmentionnées pour ce réseau de ruissellement étaient les mêmes que celles découvertes pour les bassins versant à grande échelle et pour les expérimentations en laboratoire avec des canaux observables. De plus, les lois d'échelle étaient invariantes dans le temps bien que le réseau évoluait de façon dynamique au cours de l'expérience.

La similitude statistique entre la morphologie non-canalise à petite échelle et celle à l'échelle du bassin versant nous a motivé pour tester l'applicabilité d'un modèle à base physique d'évolution des échelles du paysage (LEM) à l'échelle du laboratoire et en l'absence de ruisseau. Nous avons modélisé le ruissellement comme un réseau qui conserve le flux d'eau pour chaque cellule dans le domaine discrétisé. Ce réseau représentait l'écoulement de surface et déterminait la direction de l'évolution. Le modèle calibré prévoit une morphologie de surface plus lisse (et moins de réseaux détaillés de ruissellement) que celle mesurée puisque le sol expérimental contenait des petits cailloux qui ont émergé durant l'érosion, et qui ont été identifiés par les scans laser. Pour examiner la qualité de la modélisation, nous avons utilisé un filtre passe-bas pour éliminer la variabilité à petite-échelle de la morphologie de surface. L'analyse spectrale a confirmé que les prédictions du modèle capturent les caractéristiques principales de la morphologie mesurée. Cependant, le modèle ne peut ni reproduire les relations d'échelle expérimentales, ni la micro-rugosité de surface. Pour modifier le modèle LEM, ceci a été résolu par une équation différentiel stochastique. Il a été constaté qu'un terme additionnel pour la rugosité était nécessaire pour capturer plus de détails morphologiques. En outre, une nouvelle approche déterministe (coefficient de diffusion en tant que fonction de

courbure) a été proposée et testée pour améliorer les prédictions de modèles dans un sens statistique.

Ce travail a montré que les boucles d'hystérésis pour l'éclaboussement dominaient l'érosion et les comportements hystérétiques différents pour les différentes tailles de particule. Les métriques statistiques avantageuses (lois d'échelles) ont aussi été trouvées pour les évolutions morphologiques non-canalises et l'applicabilité du modèle LEM a été démontré en l'absence de ruisseaux. En plus des contributions scientifiques, des travaux de modélisation, des outils d'optimisation et d'analyse de données (programmes en C++ et Python) ont été développés pour servir à de futures études géomorphologiques.

Mots clés : Ruissellement, transport de sédiments, modèle d'évolution du paysage, Modèle d'érosion Hairsine and Rose (HR), Evolution des réseaux, analyse statistique, Simulation numérique, Analyse spectrale, Optimisation, C++, Python.

Zusammenfassung

Transportprozesse in der Umwelt hängen stark von der Gestalt der Landschaft ab. Moderne Einzugsgebietsanalysen mit hochauflösenden Daten und grossen Rechenleistungen verlangen nach Erosionsmodellen mit detaillierteren Metriken auf Zellenebene. Diese Dissertation beinhaltet Laborexperimente und numerische Simulationen von

niederschlagsbedingtem Sedimenttransport. Die betrachteten Grössenordnungen waren dabei deutlich kleiner, als die in der klassischen Einzugsgebietsanalyse üblicherweise verwendeten Rechenzellen. Das Hauptziel war eine stochastische und physikalische Studie der Morphologie von freiem (nicht-kanalisiertem) Oberflächenabfluss.

Im ersten Schritt wurde eine detaillierte experimentelle Studie durchgeführt, um den Einfluss von morphologischen Veränderungen auf hysteretischen Sedimenttransport unter veränderlichem Niederschlag zu untersuchen. Ein sequentielles Niederschlagsmuster, bestehend aus sieben schrittweise variierenden Niederschlagsintensitäten, wurde auf einen 5-m×2-m grossen Versuchsaufbau zur Untersuchung von Bodenerosion beaufschlagt. Es wurden im Uhrzeigersinn verlaufende Hystereseschleifen

der Sedimentkonzentration als Funktion des Abflusses gemessen, sowohl für den insgesamt abgetragenen Boden, als auch für die feineren Partikel. Für grössere Partikel nahmen Hystereseeffekte ab und die Schwebstoffkonzentrationen veränderten sich tendenziell eher linear mit dem Abfluss. Hinsichtlich des insgesamt abgetragenen Bodens, sowie für die feinen Partikel, stimmte das Hairsine und Rose (HR) Erosionsmodell gut mit den experimentellen Ergebnissen überein. Obwohl der direkte Modelabgleich für die grössten Partikel mangelhaft war, lieferte das Model eine qualitativ hinreichend gute Übereinstimmung für grössere Partikel. Grundsätzlich konnten wir zeigen, dass das Hystereseverhalten von der Partikelgrösse abhängt und

vom HR Model, innerhalb der verschiedenen Klassen von Partikelgrößen, konsistent wiedergegeben wird.

Nach der Darlegung der Rolle von morphologischen Veränderungen (Bildung einer Schutzschicht im oberen Bereich des Bodens) auf das Erosionsmuster, wurde die Morphologie des Oberflächenabflusses statistisch charakterisiert. Dazu wurden grossskalige Einzugsgebiets-Netzwerkanalysen für die Untersuchung der Mikro-Rauigkeiten von nicht-kanalisierten Oberflächen im Labormassstab angewandt (2-m × 1-m Versuchsaufbau). Die Skalierungsbeziehungen zwischen der Fläche des Einzugsgebiets und der Gewässerlänge (Hacksches Gesetz), sowie für Überschreitungswahrscheinlichkeit, Abflussmenge und stromaufwärts gelegene Netzwerklänge, sind für kanalisierte, flussartige Regionen hinlänglich bekannt. Diese Arbeit befasst sich mit der Erosion von unbefestigtem Sediment ohne eine auftretende Kanalisierung. Die morphologische Entwicklung des Sediments wurde mittels Laser-Scanning erfasst. Anhand der Laser-Scans und der Niederschlagsverteilung wurde der Oberflächenabfluss mit Hilfe des D8 Algorithmus abgeschätzt. Dieser Algorithmus lieferte ein Abfluss-Netzwerk welches statistisch untersucht wurde. Die oben erwähnten Skalierungsbeziehungen für das betrachtete Oberflächenabfluss-Netzwerk waren dieselben, wie sie für grossskalige Einzugsgebiete sowie für Laborversuche mit sichtbarer Kanalisierung berichtet wurden. Zusätzlich waren die Skalierungsgesetze zeitlich unveränderlich, obwohl das Netzwerk sich im Verlauf des Experiments dynamisch veränderte.

Die statistische Ähnlichkeit einer kleinskaligen, nicht-kanalisierten Morphologie zu Einzugsgebieten veranlasste uns dazu, die Anwendbarkeit eines physikalisch motivierten Landschafts-Entwicklungsmodells (LEM), welches üblicherweise zur Modellierung grossskaliger Einzugsgebiete verwendet wird, für unseren kleinskaligen Versuchsaufbau ohne Oberflächenrillen zu testen. Der Oberflächenabfluss wurde als ein Netzwerk modelliert, welches den (Wasser-)Abfluss für jede Rechenzelle im diskretisierten Gebiet erhält. Dieses Netzwerk stellte den Oberflächenabfluss dar und bestimmte die Evolutionsrichtung. Das kalibrierte Model sagte eine glattere Oberflächenmorphologie (und entsprechend ein weniger detailliertes Abflussnetzwerk) voraus, als experimentell gemessen wurde. Dies liegt darin begründet, dass der Boden im Experiment kleine Kieselsteine enthielt, welche während der Erosion aufgetaucht sind und welche vom Laserscanner erfasst wurden. Um die Qualität der Modelvorhersagen zu untersuchen, wurden kleinskalige Variationen in der Oberflächenmorphologie mittels eines Tiefpassfilters entfernt.

Eine Spektralanalyse bestätigte, dass das Model die Hauptcharakteristiken der gemessenen Morphologie wiedergeben kann. Die experimentell bestimmten Skalierungsbeziehungen konnte das Model hingegen nicht reproduzieren, da die Mikrorauigkeit der Oberfläche im Model nicht erfasst wurde. Um das LEM zu verbessern, wurde eine stochastische partielle Differentialgleichung gelöst. Es stellte sich heraus, dass ein zusätzlicher Term für die Rauigkeit notwendig war, um weitere Details der Morphologie zu erfassen. Des Weiteren wurde ein neuer deterministischer Ansatz (Diffusionskoeffizient als eine Stufenfunktion der Krümmung) eingeführt und getestet um die Modelvorhersagen in statistischem Sinne zu verbessern.

Diese Arbeit hat Hystereseschleifen für Tropfenaufschlag-dominierte Erosion sowie das unterschiedliche Hystereseverhalten in Abhängigkeit der Partikelgrösse gezeigt. Des Weiteren wurden sinnvolle statistische Metriken (Skalierungsgesetze) für nicht-kanalisierte Morphologie-Entwicklung gefunden sowie die Anwendbarkeit einer LEM in Abwesenheit von Oberflächenrillen gezeigt. Neben den wissenschaftlichen Beiträgen wurden nützliche Modellierungs-, Optimierungs- und Datenanalysetools (C++ und Python Codes) für folgende geomorphologischen Untersuchungen entwickelt.

Schlüsselwörter: Oberflächenabfluss, Sedimenttransport, Landschafts-Entwicklungsmodell, Hairsine und Rose (HR) Erosionsmodell, Netzwerkentwicklung, Statistische Analyse, Numerische Simulation, Spektralanalyse, Optimierung, C++, Python.

Contents

Acknowledgements	i
Abstract (English/Persian/German/French)	iii
Contents	xix
List of figures	xxi
List of tables	xxvii
List of scripts	xxix
List of Symbols	xxxi
1 Introduction	1
1.1 Environmental impacts of erosion	1
1.2 Research motivation	1
1.3 Theories and concepts	2
1.3.1 Hairsine-Rose erosion model	2
1.3.2 Catchment scaling laws	4
1.3.3 The physically-based landscape evolution model	4
1.4 State of the art	8
1.4.1 Geometrical characteristics of landscapes	8
1.4.2 Laboratory-scale rain-induced sediment transport	11
1.5 Research questions and methodologies	29
2 Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects	31
2.1 Introduction	32
2.2 Methods	36

Contents

2.2.1	Experiment	36
2.2.2	Model application and parameter estimation	38
2.3	Results and discussion	39
2.3.1	Experiment	39
2.3.2	Model	49
2.4	Concluding remarks	52
3	Statistical characteristics of overland flow morphologies	55
3.1	Introduction	56
3.2	Experiment	57
3.3	Results and discussion	59
3.3.1	Morphology and drainage network evolution	59
3.3.2	Catchment scaling laws in overland flow morphologies	62
3.3.3	Further morphological features	68
3.3.4	Another experiment	71
3.4	Physical interpretation of the invariant scaling laws	76
3.5	Conclusions	77
4	Landscape evolution model in absence of rills	79
4.1	Introduction	80
4.2	Numerical simulation and calibration	83
4.3	Model parameter study	85
4.4	Results of the original model (uniform parameters)	87
4.4.1	Morphology	87
4.4.2	Spectral analysis and filtering	89
4.4.3	Discharge drainage network	95
4.4.4	Statistical analysis	97
4.5	LEM modification to enhance the statistical features	101
4.5.1	Spatially non-uniform diffusion based on precipitation distribution (Model 2)	101
4.5.2	LEM as a stochastic partial differential equation (Models 3-7) . .	101
4.5.3	Diffusion control via a critical curvature (Model 8)	106
4.5.4	Parameter optimization of the modified models	107
4.6	Results of the modified models	108
4.7	Conclusions	124

5	Conclusions and outlook	127
5.1	Summary and conclusions	127
5.1.1	Rainfall-driven hysteretic sediment fluxes as a result of morphological changes	127
5.1.2	Statistical characteristics of unchanneled morphology evolution	128
5.1.3	Catchment-scale LEM on unchanneled morphology evolution .	128
5.2	Outlook	129
	Data availability statement	131
A	More details of the experimental setups	133
A.1	Surface scanning and image registration	133
A.2	Rainfall distribution measurement	135
A.3	Sediment selection for the experiment	135
B	Model verification	139
B.1	1-D advection-diffusion	139
B.2	2-D diffusion	140
C	C++ scripts: Landscape evolution model and optimization tools	143
D	Selected Python scripts: Data analysis and visualization	193
	Bibliography	243
	Curriculum Vitae	245

List of Figures

2.1	Schematic of the EPFL soil erosion flume. The flume slope can be varied between 0 and 30%. Precipitation is applied using 10 oscillating sprinklers located 3-m above the soil surface.	37
2.2	Seven sequential rainfall events, each 20-mins long.	38
2.3	Discharge measured at the two collectors. Except the first rainfall event (precipitation onto a dry soil), the discharge rapidly adapts to changes in rainfall intensity.	40
2.4	Total sediment concentration (C) and the concentration of the seven different particle size classes ($C_1 - C_7$) for Collector 1. Regardless of the rainfall intensity, concentrations of the first two size classes decrease with time whereas concentrations of the larger particle sizes vary with the rainfall intensity.	42
2.5	Total sediment concentration (C) and concentrations of the seven different particle size classes ($C_1 - C_7$) for Collector 2. The HR model predicts the concentrations of the finer particle size classes satisfactorily. As the particles become larger, the difference between the model predictions and experimental data increases.	43
2.6	Hysteresis loops (measured data and HR model results) associated with the total sediment concentration (C) and individual particle size classes ($C_1 - C_7$) (Collector 1). The elapsed time (0-140 min) is shown by the color bar.	46
2.7	Hysteresis loops (measured data and HR model results) associated with the total sediment concentration (C) and individual particle size classes ($C_1 - C_7$) (Collector 2). The elapsed time (0-140 min) is shown by the color bar.	47

List of Figures

2.8	Schematic depiction of raindrop impact on the soil surface. Finer particles are available for suspension due to raindrop impact on the rising limb. Over time, the finer particles are preferentially removed, leaving a shield layer composed mainly of larger particle sizes [Colby, 1963].	48
2.9	HR model predictions of the total mass of the deposited layer (m) and the contribution of seven different particle size classes ($m_1 - m_7$) within the shield layer (Collector 1). The variation of degree of shielding (H) is shown along with the total mass.	51
3.1	Schematic of the flume experiment. The flume width, W , was 1 m and its length, L , was 2 m. The sediment vertical elevation (z) was measured from the outlet ($z = 0$). The relief height (the maximum elevation) was 0.19 m. Each layer was compacted by 30-min rainfall (droplet size of 3-7 mm) with an intensity of 10 mm h^{-1} followed by consolidation via a 600 kg m^{-2} weight. Then, the sediment was air dried for 48 h.	58
3.2	Particle size distribution of the sediment.	59
3.3	Measured water flow rate at the flume outlet.	60
3.4	Measured morphology evolution during the 16-h experiment. Due to the non-uniform rainfall, the morphology changes increase from the left side (low precipitation rate area) towards the right (high precipitation rate area).	61
3.5	Drainage area (A) distribution determined using the D8 algorithm and the measured morphologies shown in Figure 3.4a-g. Initially, the flow paths, e.g., at $t = 0.25$ and 0.5 h , reflect the initial surface condition and central drainage point at the flume exit. The impact of the higher-intensity rainfall on the right side of the flume is manifested in the main flow path, which increasingly moves to the right side over time (more details given in the text).	63
3.6	Drainage discharge distribution determined using the D8 algorithm and the measured morphologies and the rainfall intensity (Figure 3.4). In spite of the heterogeneous rainfall intensity, the generated network and its evolution are very similar to the drainage area network (Figure 3.5). As expected, the maximum calculated discharge at the outlet (2.709 L min^{-1}) is consistent with the measured value at the steady state (Figure 3.3). .	64

3.7	Width function (w) of the drainage area (and discharge) network (Figures 3.6 and 3.5). The value X is the distance of each point from the flume outlet (along the flow paths of the network) and $w(X)$ is the number of points with the same distance (X) from the network outlet. The width function of the network dynamically changed during the morphology evolution.	65
3.8	Sediment surface at $t = 0$ (a) and $t = 16$ h (b).	66
3.9	Relation between the upstream length and drainage area ($l = A^h$). The curves of higher moments ($n > 1$) are shifted vertically for the purpose of visualization.	67
3.10	Exceedance probabilities of discharge (Q), drainage area (A) and upstream length (l) at different times.	68
3.11	Slope versus drainage area. A power law describes the intermediate drainage areas.	69
3.12	Curvature versus the product of drainage area and slope. The average values of curvature are negative where the product of area and slope ($A \nabla z $) is low whereas positive curvature values are observed as $A \nabla z $ increases.	70
3.13	Elevation field at 10% slope and 60 mm h^{-1} average rainfall.	72
3.14	Drainage area at 10% slope and 60 mm h^{-1} average rainfall.	73
3.15	Discharge flux at 10% slope and 60 mm h^{-1} average rainfall.	74
3.16	Plot of Hack's law ($l = A^\alpha$) at 10% slope and 60 mm h^{-1} average rainfall. The curves of higher moments ($n > 1$) are shifted for the purpose of visualization.	75
3.17	Scaling relation for the distribution of discharge (a), drainage area (b) and length (c) at 10% slope and 60 mm h^{-1} average rainfall.	75
4.1	The wall boundary condition assuming \vec{n} as the unit vector normal to walls in outward direction and dn is the discretization size along \vec{n} . $\frac{\partial z^e}{\partial n} _w^t$ was calculated by linear interpolation between the two consecutive scans before and after the time t	84
4.2	Model results after 1 h for $K = 0.2 \text{ mm}^{-\frac{1}{2}} \text{ h}^{-\frac{1}{2}}$, $\Omega_{cr} = 10 \text{ mm}^{\frac{3}{2}} \text{ h}^{-\frac{1}{2}}$ and $D (\text{mm}^2 \text{ h}^{-1}) = 0$ (a), 250 (b), 500 (c), 1000 (d), 2000 (e), 4000 (f), 8000 (g), 16000 (h). The incision rate of the surface decreases as the diffusion coefficient (D) increases.	86

List of Figures

4.3	Numerical simulation of the LEM. For comparison, the experimental data (Figure 3.4) are repeated in this figure.	88
4.4	2-D power spectral density of the experimental and model results. Low wave numbers are more in agreement.	91
4.5	Power spectral density of the morphology. The errorbars of the experimental data are based on the standard deviation from the azimuthally averaged PSD in the wave number domain.	92
4.6	The original and low-pass filtered morphologies at $t = 8$ h.	93
4.7	The original discharge network and the network extracted from the low-pass filtered morphology at $t = 8$ h.	94
4.8	Discharge distribution at different times. The experimental network was calculated based on the filtered morphology.	96
4.9	The relation between the upstream length and drainage area (Hack's law) for the model and experiment. The model is able to reproduce the filtered experimental data.	98
4.10	Exceedance probability of drainage area for the model and experiment. The model is able to reproduce the filtered experimental data.	99
4.11	Exceedance probability of upstream length for the model and experiment. The model is able to reproduce the filtered experimental data.	100
4.12	The roughness distribution at different times ($t = 0.25, 0.5, \dots, 16$ h). Roughness is defined as the difference between morphology and its filtered value using the Blackman window (section 4.4.2).	103
4.13	The probability distribution of roughness during the 16-h course of the experiment. The outliers (more than 95% percentile) are removed from the dataset, and the values are normalized based on the maximum roughness.	104
4.14	The probability distribution functions that were calibrated with the roughness distribution in Figure 4.13. The PDF numbers (1-77) are assigned based on their fitness with the experimental roughness (error value in the plot). The error value is the root mean square error between the experimental distribution and calibrated PDF. The distribution names are taken from the Python library.	105
4.15	Calculation of the diffusion coefficient at each time step, after solving the advection term and before applying the diffusion.	107

4.16 Elevation field (z) of the models at $t = 8$ h. Model 1 is the original LEM (equation 1.16). In Model 2, the diffusion coefficient is correlated with the rainfall distribution. Model 3 to 7 are the stochastic PDEs which are defined in Table 4.2. In Model 8, the diffusion coefficient is a function of curvature (equation 4.5).	109
4.17 Power spectral density of the elevation fields (z) at $t = 8$ h at the logarithmic scale. Model 1 is the original LEM and Models 2 to 8 are defined in section 4.5. The low-wave number PSD values of the models and experiment are in the same range whereas the models behave differently at high wave numbers.	112
4.18 The azimuthally-averaged power spectral density of the morphology in the wave number domain. The errorbars of the experimental data are based on the standard deviation from the mean. Model 1 is the original LEM and Models 2 to 8 are defined in section 4.5.	113
4.19 Discharge distribution of the experiment and different models at $t = 8$ h. Model 1 is the original LEM and Models 2 to 8 are defined in section 4.5.	114
4.20 Continued	116
4.20 Elevation field of the experiment and Models 1, 2 and 8. Model 1 is the original LEM and Models 2 and 8 are the modifications defined in section 4.5.	117
4.21 Continued.	118
4.21 Discharge distribution of the experiment and Models during the 16-h course of experiment. Compared to the original LEM, Models 6 and 8 are more in agreement with the experiment by producing a more branched network at the low-discharge.	119
4.22 The relation between the upstream length and drainage area (Hack's law) for the models and experiment. Models 6 and 8 improve the original LEM to follow Hack's law. However, Model 8 shows more enhancement compared to Model 6.	120
4.23 Exceedance probability of drainage discharge for the experiment and Models 1, 6 and 8.	121
4.24 Exceedance probability of drainage area for the experiment and Models 1, 6 and 8.	122
4.25 Exceedance probability of upstream length for the models and experiment.	123

List of Figures

A.1	The pieces of scans on the flume. The numbers (1-28) show the order of scanning. The pieces were registered using the intersection areas (a_1 , $a_2 \dots c_2$). Four points A, B, C, D were the labels on the flume's frame to coordinate the overall geometry.	134
A.2	Digital Elevation Model of the flume. The 28 pieces of the 120cm-WIDE case were joined by translation along the width and registration along the flume's length. All values are in (mm).	135
A.3	Rainfall distribution measurement on the flume.	136
A.4	Morphology evolution of experiment 6 (Table A.2).	138
B.1	Analytical and numerical solution of equation B.1 with $D, K, A, \alpha = 1$. .	141
B.2	Numerical and analytical results of equation B.4 with $\psi = 1$: (a-d) Numerical results at the initial times, (e) numerical and (f) analytical (equation B.6) steady state solutions.	142

List of Tables

1.1	Selected works on geometrical characteristics of landscapes.	8
1.2	Some laboratory works on single drop impact on granular beds.	12
1.3	Laboratory-scale studies of rainfall driven sediment transport on unchanneled surfaces.	16
1.4	Experimental investigations of sediment transport and network analysis on incised beds.	19
2.1	Properties of seven different particle size classes of the soil. The settling velocities are from <i>Tromp-van Meerveld et al.</i> [2008].	37
2.2	Optimized Parameters (a , a_d , m^* and D) for each rainfall event (E1-E7).	39
3.1	Characteristics of the sand used in the experiment.	58
4.1	Calibrated parameters for the original LEM (Model 1).	87
4.2	Model assumptions for different stochastic PDEs. The parameter η is a random variable from the Half-Cauchy probability distribution function (equation 4.3).	106
4.3	Calibrated parameters for Model 2. The diffusion coefficient at each cell ($D(x, y)$) is proportional to the rainfall intensity, $R^*(x, y)$. The spatially averaged diffusion coefficient is shown by \bar{D}	110
4.4	Calibrated parameters for the stochastic PDEs (Models 3-7). The spatially random coefficients are defined according to Table 4.2.	110
4.5	Optimized parameters for Model 8. The diffusion coefficient is defined as a step function of curvature (equation 4.5, Figure 4.15).	111
A.1	Dimension and resolution of the scanner with different lens types at various distances from the surface.	134

List of Tables

A.2 The experiments carried out to achieve an unchanneled fast evolving morphology. Different rainfall intensities, sediment types and flume sizes were used. 137

Listings

C.1	'header.cpp': List of functions and their description	143
C.2	'main.cpp': The main functions	145
C.3	'network.cpp': Drainage network extraction	149
C.4	'numeric.cpp': Matrix Operation and writing data	158
C.5	'RUN.cpp': Numerical modeling	165
C.6	'optimization.cpp': Different objective functions and optimisation . . .	167
C.7	'analysis.cpp': Network analysis and data extraction	175
C.8	'matrixOperation.cpp': Basic matrix operations	183
C.9	'K_means.cpp': 1D and 2D K-means algorithm	186
C.10	'LANDserial.cpp': Coupling the model with multi-objective Borg optimization code	189
D.1	Exceedance probability of lenght	193
D.2	Data analysis and visualization	194

Nomenclature

\bar{D}	Mean diffusion coefficient ($L^2 T^{-1}$)
β	Scaling exponent for the exceedance probability of drainage area (–)
η	Random value from the fitted distribution (–)
Ω	Flow stream power per unit area ($M L^{-1} T^{-2}$)
Ω_{cr}^*	Critical stream power per unit area ($M L^{-1} T^{-2}$)
Ω_{cr}	Adjusted critical stream power in the LEM ($L^{\frac{3}{2}} T^{-\frac{1}{2}}$)
Ω_{cr}^{max}	Maximum critical stream power in the modified LEM ($L^{\frac{3}{2}} T^{-\frac{1}{2}}$)
ϕ	Diffusion coefficient at high curvatures ($L^2 T^{-1}$)
ψ	Scaling exponent for the exceedance probability of upstream length (–)
ρ_r	Rock bulk density ($M L^{-3}$)
ρ_s	Sediment bulk density ($M L^{-3}$)
ρ_w	Water density ($M L^{-3}$)
$\vec{\tau}$	Flow shear stress flux ($M L^{-2} T^{-1}$)
\vec{S}	Slope vector (–)
\vec{V}	Velocity vector ($L T^{-1}$)
A	Drainage area (L^2)
a	Detachability of the original soil ($M L^{-3}$)
a_d	Detachability of the deposited soil ($M L^{-3}$)

Listings

C_i	Suspended sediment concentration of particle size i (ML^{-3})
C_{cr}	Critical curvature (L^{-1})
D	Diffusion coefficient in the LEM ($\text{L}^2 \text{T}^{-1}$)
D	Surface water depth in the HR model (L)
D_{max}	Maximum diffusion coefficient in the modified LEM ($\text{L}^2 \text{T}^{-1}$)
$H = m_t / m^*$	Degree of shielding (–)
h	Hack's law exponent (–)
K	Advection coefficient in the LEM ($\text{L}^{-\frac{1}{2}} \text{T}^{-\frac{1}{2}}$)
k_1	Empirical parameter defining the sediment transportation capacity (T M^{-1})
k_2	Empirical parameter in the discharge-velocity relation (–)
K_{max}	Maximum advection coefficient in the modified LEM ($\text{L}^{-\frac{1}{2}} \text{T}^{-\frac{1}{2}}$)
l	Upstream length (L)
M	Roughness coefficient ($\text{L}^2 \text{T}^{-1}$)
m	Discharge exponent in the LEM (–)
m^*	Required deposition mass per unit area for a complete shielding of the original soil (ML^{-2})
m_i	Deposited sediment mass per unit area (ML^{-2})
$m_t = \sum m_i$	Total mass of the deposited layer per unit area (ML^{-2})
N	Number of particle sizes in the HR model (–)
P	Rainfall rates in the HR model (L T^{-1})
$P(A > a)$	Exceedance probability of drainage area (–)
$P(L > l)$	Exceedance probability of upstream length (–)
p_i	Mass proportion of the particle size class i in the original soil (–)

Q	Discharge ($\text{L}^3 \text{T}^{-1}$)
q	Volumetric water flux per unit width ($\text{L}^2 \text{T}^{-1}$)
q_c	Volumetric flux per unit width of shear-driven sediment transport ($\text{L}^2 \text{T}^{-1}$)
q_m	Volumetric flux per unit width of mass-movement sediment transport ($\text{L}^2 \text{T}^{-1}$)
q_s	Volumetric total sediment flux per unit width ($\text{L}^2 \text{T}^{-1}$)
R	Excess rainfall in the HR model (L T^{-1})
R	Hydraulic diameter in open channel flow (L)
$R^*(x, y)$	Normalized (based on the mean) rainfall distribution (—)
t	Time (T)
U	Uplift rate (M L^{-1})
v_i	Settling velocity (L T^{-1})
w	Open channel width (L)
x	Downslope distance in the HR model (L)
x	Transversal direction of the flume for the LEM simulations (L)
y	Longitudinal direction of the flume for the LEM simulations (L)
$z(x, y, t)$	Elevation (L)
$z^*(x, y, t)$	Elevation after solving the advective term of the LEM (L)

1 Introduction

1.1 Environmental impacts of erosion

Sediment transport is a significant environmental and agricultural problem that causes numerous disastrous side effects such as the removal of fertile agrarian soil [Pimentel *et al.*, 1995; Tilman *et al.*, 2002; Novara *et al.*, 2018], the release of carbon from the lithosphere into the atmosphere [Quinton *et al.*, 2010; Malusà *et al.*, 2018] and the mobilization of pollutants from the ground into water systems [Pavlidis and Tsihrintzis, 2018]. The recent development of industrial activities and road constructions has intensified this process despite the dramatically increased demand for food and clean drinking water as a result of the global increase in population size. In general, water and wind erosion impact 56 and 28 percent, respectively of degraded lands and the overall cost of soil erosion in the world is estimated to be 400 billion US dollars annually [Blanco-Canqui and Lal, 2010]. This cost includes the money spent on the renewing of nutrients, cleaning water resources and preventing erosion. Thus, research, management, and conservation of soil resources have become more critical in recent years [Rodrigo-Comino, 2018].

1.2 Research motivation

In general, most of the shortcomings and failures of the current erosion models refer to the fact that a static initial surface is assumed in the models and erosion is calculated based on that surface, while there is a dynamic relationship between morphological changes of a surface and erosion rate. In addition to the soil

Chapter 1. Introduction

transport estimation accuracy, the land's shape provides the geometry for transport processes. Therefore, a more detailed geometrical characterization of the landscape is crucial for soil management and different environmental analysis of water systems. This dissertation involves detailed experiments and numerical simulations of rainfall-driven morphological evolution at the laboratory scale and under controlled conditions. The flumes' areas are smaller than the common computational cell sizes at the catchment scale. The idea is to analyze the morphological evolution of unchanneled surfaces both statistically and physically. Detailed research questions and methodologies are presented in section 1.5.

1.3 Theories and concepts

1.3.1 Hairsine-Rose erosion model

In contrast to wind erosion, water erosion exists in different climates and is the dominant factor of erosion. This process is quite complex [Herman *et al.*, 2015] and, depending on conditions, different types of splash, rill, interrill, gully, tunnel and streambank erosion can occur [Blanco-Canqui and Lal, 2010; Morgan and Nearing, 2011]. Several models were proposed for soil erosion prediction and sediment transport characterization at both field and laboratory scales [Morgan and Nearing, 2011]. In large-scale models, the overall behavior of the erosion phenomenon is measured, while process-based models focus mainly on the underlying detailed physical descriptions. Different process-based models have been proposed by researchers such as Hairsine-Rose (HR) model [Hairsine and Rose, 1991, 1992a, b], Western European Prediction Erosion Project (WEPP) model [Flanagan and Nearing, 1995; Flanagan *et al.*, 2001], Kinematic Runoff Erosion (KINEROS) model [Smith *et al.*, 1995], Limburg Soil Erosion Model (LISEM) [De Roo *et al.*, 1996] and European Soil Erosion Model (EUROSEM) [Morgan *et al.*, 1998]. Among these models, the HR model provides a unique description of erosion that includes different particle sizes with different settling velocities and generation of a shield layer on the soil, which causes protection of the original soil [Sander *et al.*, 2011].

The HR model considers different particle size classes and incorporates a mechanistic description of a shield layer development (that is composed of previously eroded material that helps to protect the original soil from further erosion). This model

was investigated theoretically [Sander *et al.*, 1996; Lisle *et al.*, 1998; Hairsine *et al.*, 1999; Parlange *et al.*, 1999; Barry *et al.*, 2010; Kinnell, 2013] and validated via zero-dimensional [Heilig *et al.*, 2001; Gao *et al.*, 2003] and flume-scale laboratory experiments [e.g., Jomaa *et al.*, 2010, 2012b, 2013], as well as at the small field [Van Oost *et al.*, 2004]. Details of the HR model are available elsewhere [Hairsine and Rose, 1991, 1992a, b], so only a summary is given here. The model's governing equations in the absence of flow-driven processes are:

$$\frac{\partial D}{\partial t} + \frac{\partial q}{\partial x} = R \quad (1.1)$$

$$\frac{\partial(D C_i)}{\partial t} + \frac{\partial(q C_i)}{\partial x} = a p_i P(1 - H) + \frac{m_i}{m_t} a_d P H - v_i C_i, \quad i = 1, 2, \dots, N \quad (1.2)$$

$$\frac{\partial m_i}{\partial t} = v_i C_i - \frac{m_i}{m_t} a_d P H \quad i = 1, 2, \dots, N. \quad (1.3)$$

where i refers to particle size class, N is the number of size classes, t , x , D , and C_i denote the time (s), down-slope distance (m), surface water depth (m), and suspended sediment concentration (kg m^{-3}), respectively, m_i is the deposited sediment mass per unit area (kg m^{-2}), q is the volumetric water flux per unit width ($\text{m}^2 \text{s}^{-1}$), R and P are the excess rainfall and rainfall rates (m s^{-1}), respectively, a is the detachability of the original soil (kg m^{-3}), p_i is the mass proportion of the particle size class i in the original soil, a_d is the detachability of the deposited soil (kg m^{-3}), and v_i is the settling velocity (m s^{-1}). The degree of shielding is measured by $H = m_t / m^*$ in which $m_t = \sum m_i$ is the total mass of the deposited layer and m^* is the required mass for a complete shielding of the original soil.

In this dissertation, the HR model is calibrated with the experimental data during a time-varying precipitation rate (Chapter 2). The calibrated model provided the physical support for the importance of near-surface particle rearrangements on erosion patterns of seven different particle size classes of the soil.

1.3.2 Catchment scaling laws

Landscapes are generally separated into hillslopes and fluvial sections (river networks). In spite of their different climates, soil types and scales, river networks have common geometric characteristics which were widely investigated in the literature. For example, Hack's law [Hack, 1957] defines a scaling relation between the upstream length (l , the longest flow path into each point) and drainage area (A):

$$l = A^h \quad (1.4)$$

where the exponent h was measured in the range of [0.5-0.7] for different river networks [Hack, 1957; Gray, 1961; Mueller, 1972; Rigon *et al.*, 1996, 1998]. Moreover, for the fluvial parts of landscapes, power-law relations were found for the exceedance probability of drainage area ($P(A > a)$) and upstream length ($P(L > l)$):

$$P(A > a) = a^{-\beta} \quad (1.5)$$

$$P(L > l) = l^{-\psi} \quad (1.6)$$

where the exponents β and ψ are in the ranges of [0.42-0.45] and [0.5-0.9], respectively [Rodriguez-Iturbe and Rinaldo, 1997; Rigon *et al.*, 1996; Crave and Davy, 1997; Paik and Kumar, 2011]. More investigations on landscape geometric features are reviewed in section 1.4.1. In this dissertation, we use these metrics to characterize unchanneled areas by considering high-resolution morphologies of overland flows at the laboratory scale (Chapter 3).

1.3.3 The physically-based landscape evolution model

Following Gilbert's report [Gilbert, 1877], descriptive models for river networks emerged [e.g., Gilbert, 1877; Davis, 1892; Glock, 1931; Horton, 1945; Woldenberg, 1966]. One of the first quantitative physical descriptions of landscape evolution was given by Davis [1892]: "The form assumed by the surface of the land depends largely on the ratio between the processes of washing and creeping". Afterwards, fundamental studies on the mathematical formulation of these two complicated

processes (“creeping” and “washing”) continued until 1989 [e.g., *Smith and Bretherton*, 1972; *Carson and Kirkby*, 1972; *Kirkby*, 1967, 1971, 1985; *Ahnert*, 1976, 1977, 1987; *Dunne and Aubry*, 1986]. These studies mostly concentrated on either hillslopes [e.g., *Kirkby*, 1971; *Dunne*, 1980] or fluvial networks [e.g., *Shreve*, 1967, 1969; *Mock*, 1971] and the coupling between these two regions was not robustly modeled via physical laws. In 1989, Willgoose comprehensively classified the investigations on landscape evolution in his Ph.D. thesis [*Willgoose*, 1989] and introduced the first physically-based 2-D model with high-resolution numerical simulation [*Willgoose et al.*, 1991, 1992]. The model was inspired by a non-hydrologic network model that simulated the growth of leaf veins [*Meinhardt*, 1976, 1982] where the process within each cell was defined via an activation equation. Then, *Howard et al.* [1994] introduced a new model without an activation parameter. With this model, it was possible to have both creeping and flow-driven erosion within one cell. The model was followed by Tucker and Slingerland’s numerical simulation [*Tucker and Slingerland*, 1994] in which they took into account bedrock weathering [*Ahnert*, 1976] and bedrock channel incision [*Seidl and Dietrich*, 1992; *Howard*, 1987; *Howard et al.*, 1994] phenomena. The landscape evolution model (LEM) used in this work, was firstly introduced by *Howard et al.* [1994], which was then further extended and numerically solved by *Perron et al.* [2008]. The derivation below is the same as *Perron et al.* [2008] with the only difference that here we assume a critical stream power rather than critical shear stress for the onset of rill erosion. By defining $z(x, y, t)$ as surface elevation from a fixed base level, the LEM stems from the conservation of mass:

$$\rho_s \left(\frac{\partial z}{\partial t} + \nabla \cdot \vec{q}_s \right) = \rho_r U \quad (1.7)$$

where t is time, ρ_s and ρ_r are respectively the bulk density of sediments and rock, and U is the uplift rate ($U = 0$ in our analysis). The parameter, q_s , is the volume flux of transportable sediment per unit width of the land surface rate which its gradient consists of two components:

$$\nabla \cdot \vec{q}_s = \nabla \cdot \vec{q}_m + \nabla \cdot \vec{q}_c \quad (1.8)$$

where q_c is the sediment transportation associated with water flow within the channels and q_m represents the mass sediment transport. The mass sediment transport stems from a variety of processes such as rain splash, creeping and frost harvesting, and

Chapter 1. Introduction

drying-wetting cycles. In our experiment, the main effects are rain splash and creeping. This term can be modeled as a flux that is proportional to the topographic gradient with the coefficient D :

$$\vec{q}_m = -D \nabla z \quad (1.9)$$

The flow-driven sediment transport is caused by the fluid flow within rivers. This term can be calculated based on two criteria: I) stream power and II) shear stress. By defining Ω as the flow stream power and Ω_{cr}^* as the critical stream power, the flow-driven sediment transport flux is:

$$\nabla \cdot \vec{q}_c = -k_1 f(\Omega - \Omega_{cr}^*) \quad (1.10)$$

where the coefficient k_1 is an empirical parameter which determines sediment transportation capacity and $f(\zeta) = \zeta H(\zeta)$ with $H(\zeta)$ being the Heaviside step function. The stream power is defined as the production of shear stress and fluid velocity:

$$\Omega = \vec{\tau} \cdot \vec{V} \quad (1.11)$$

For a uniform open channel flow, the steady shear stress flux ($\vec{\tau}$) is:

$$\vec{\tau} = \rho_w g R \vec{S} \quad (1.12)$$

where ρ_w , R and S are the water density, hydraulic diameter, and surface slope, respectively. By assuming w and V as the channel width and the flow velocity, respectively, the flow rate is:

$$Q_w = k_2 R w V \quad (1.13)$$

where k_2 is an empirical coefficient. A relation defines the channel width as a function of the flow rate [*Leopold et al.*, 1953; *Rodríguez-Iturbe and Rinaldo*, 1997; *Knighton*, 2014]:

$$w = k_3 Q_w^a \quad (1.14)$$

where k_3 and a are the coefficients and Q is the flow rate. By combining the relations 1.11 to 1.14, the expression for Ω is:

$$\Omega = \frac{\rho_w g}{k_2 k_3} Q_w^{1-a} S \quad (1.15)$$

Finally, the transport equation is in the form of:

$$\frac{\partial z}{\partial t} = D \nabla^2 z - K f(Q^m S - \Omega_{cr}) \quad (1.16)$$

where

$$K = \rho_w g \frac{k_1}{k_2 k_3} \quad \Omega_{cr} = \frac{k_1}{K} \Omega_{cr}^*$$

The two parameters D and K are functions of rainfall intensity, droplet size, sediment properties (e.g., density, particle size, cohesion, etc.) and surface roughness [Furbish *et al.*, 2007; Dunne *et al.*, 2010; Mahmoodabadi and Sajjadi, 2016; Sadeghi *et al.*, 2017a]. The exponent, m , takes values in the range of 0.41-0.857 [Willgoose, 1989; Rodríguez-Iturbe and Rinaldo, 1997]. At large scales, the first term on the right-hand side of equation 1.16 is dominant on hillslopes and includes different processes such as weathering [Perron, 2017], creeping and rain splash [Culling, 1960, 1963, 1965]. In Chapter 4, the LEM (equation 1.16) is calibrated with a flume-scale experiment which includes an unchanneled overland morphology evolution under a spatially non-uniform precipitation rate. The second term on the right-hand side of the equation is usually assumed to model sediment transport within the river network (since this term vanishes elsewhere). However, in our flume-scale experiment, there are no surface incisions, and the overland flow is at all times continuous across the entire flume. Thus, the overland flow in the experiment is modeled as a network, although there is no river network, as found in previous applications of the LEM [Tucker and Hancock, 2010; Benaïchouche *et al.*, 2016; Whipple *et al.*, 2016; Perron, 2017; Hancock *et al.*, 2017]. We selected $m = 0.5$ as this value is also used in studies on optimal channel networks based on the theory of minimum energy expenditure for river network evolution [Rinaldo *et al.*, 1992, 1993].

1.4 State of the art

1.4.1 Geometrical characteristics of landscapes

River networks are analogized as veins of the Earth. Therefore, their geometrical structure and organization are among the essential factors in the environmental transport phenomena and water resource management. The selected researches of this domain are presented in Table 1.1. The works include finding both the geometrical characteristics and/or physical reasons behind their fractal shapes.

Table 1.1 – Selected works on geometrical characteristics of landscapes.

Author(s)	Remarks
<i>Glock</i> [1931]	He postulated that the network evolution includes two stages of “extension” in which the network grows and increase its complexity and “integration” where the network simplifies itself by reducing the number of branches.
<i>Horton</i> [1945]	He took the initial steps toward characterizing river network structure by calculating the ratio of stream lengths and number of streams for each order class of a river network.
<i>Hack</i> [1957]	He found that the upstream length (l , the longest flow path into each point) and drainage area (A) were related via a scaling relation ($l = A^h$). The exponent h was measured in the range of [0.5-0.7] for different river networks [<i>Gray</i> , 1961; <i>Mueller</i> , 1972; <i>Mosley and Parker</i> , 1973; <i>Montgomery and Dietrich</i> , 1992; <i>Maritan et al.</i> , 1996; <i>Rigon et al.</i> , 1996, 1998; <i>Willemin</i> , 2000; <i>Sassolas-Serrayet et al.</i> , 2018].

Continued on next page

Table 1.1 – continued from previous page

Author(s)	Remarks
<i>Mandelbrot</i> [1982]	He described the fractal features of different natural systems such as clouds, mountains, rivers and watersheds.
<i>Tarboton et al.</i> [1988]	River networks were analyzed to find out their fractal dimension using Richardson [<i>Richardson</i> , 1961] and box-counting [<i>Lovejoy et al.</i> , 1987] methods. They reported a fractal dimension of 2 for the networks.
<i>Matsushita and Ouchi</i> [1989]	An algorithm to determine the self-affine exponents of curves was introduced. This approach can be used to determine the fractal dimension of the perimeter of a basin or its main channel curve.
<i>Marani et al.</i> [1991]	They studied the fractal dimension of river networks and showed that in the limit, the bifurcation ratio equaled the total area ratio.
<i>Rinaldo et al.</i> [1992]	In this paper, they analyzed the relation between the minimum energy expenditure and fractal structure of river basins. The concept of minimal energy expenditure in river networks was introduced based on three principles: (1) minimum energy expenditure in any link of the network, (2) equal energy expenditure per unit area of channel anywhere in the network, and (3) minimum total energy expenditure in the network as a whole. It was concluded that the fractal behavior of networks was a product of the least energy expenditure.

Continued on next page

Chapter 1. Introduction

Table 1.1 – continued from previous page

Author(s)	Remarks
<i>Rinaldo et al.</i> [1993]	They introduced fractal river networks as two dimensional self-organized critical systems that adjust their structure during the random perturbations of tectonic or hydrologic origin.
<i>Helmlinger et al.</i> [1993]	They used the Digital Elevation Model (DEM) data of two river basins and studied the effect of threshold area on different features such as drainage density, number of streams, basin order, Horton's ratios and fractal dimension. They concluded that there was a considerable variation in different parameters depending on the chosen threshold area.
<i>Rigon et al.</i> [1996]	By analyzing the power laws (based on the exceedance probability of drainage area and moments of the upstream length), they found interdependence of the power law exponents.
<i>Rodriguèz-Iturbe and Rinaldo</i> [1997]	The book contains a broad description of geometrical features of river basins from basic to advanced levels.
<i>Birnir</i> [2008]	Through an in-depth mathematical analysis of river flows, he derived a theoretical value of 0.571 for Hack's exponent.
<i>Sassolas-Serrayet et al.</i> [2018]	They demonstrated that the catchment shape depends on the Hack's coefficient and not on its exponent.

1.4.2 Laboratory-scale rain-induced sediment transport

At the field scale, dynamic changes of the network structure require long time scales [Shumilovskikh *et al.*, 2016] and include many uncertain factors such as climate [e.g., Nearing *et al.*, 2004; Abbaspour *et al.*, 2009; Jafari and Bakhshandehmehr, 2013; Azari *et al.*, 2016; Cerdà *et al.*, 2018; Wu *et al.*, 2018], chemical and physical properties of sediments [e.g., Certini, 2005; Pouyat *et al.*, 2007; van der Meij *et al.*, 2018; Morvan *et al.*, 2018; Sheng *et al.*, 2018; Karaca *et al.*, 2018], episodic gully erosion [Patton and Schumm, 1975; Prosser *et al.*, 1994; Poesen *et al.*, 1998; Leenman and Tunnicliffe, 2018], different vegetation types [e.g., Zhou *et al.*, 2008; Mohammad and Adam, 2010; Zhang *et al.*, 2015; Zhou *et al.*, 2016; Feng *et al.*, 2018; Ding *et al.*, 2018], tectonic effects [e.g., Ballato *et al.*, 2015; Vanmaercke *et al.*, 2017; Jaber *et al.*, 2018], land use [e.g., Yang *et al.*, 2003; Khalili Moghadam *et al.*, 2015; Zare *et al.*, 2017] and the rate of sediment production [Jaafari *et al.*, 2015; Naghdi *et al.*, 2017]. Experimental setups are extremely useful tools to understand the physics and further model development. In this section, highlighted experimental works on sediment transport and morphology evolution during a simulated rainfall are reviewed. The experiments are separated into three main categories: I) single drop impact (Table 1.2), II) unchanneled surfaces (Table 1.3) and III) channeled morphologies (Table 1.4).

1.4.2.1 Single drop impact on granular beds

This thesis includes flume-scale investigations on the rainfall-driven morphological evolution which is the accumulation of grain-scale changes on the surface during the rainfall event. Therefore, to have a comprehensive review of the experimental works, single drop impact studies are included here. Raindrop impacts are the first interactions of land and water which different kinds of processes and surface morphologies can result depending on droplet momentum, particle distribution of the granular bed, cohesion and humidity. This phenomenon is quite complex and involves many unknown parameters. In recent years, particle-based models were developed to directly simulate processes that involve fluid-particle and particle-particle interactions via coupling the Navier-Stokes equations and granular models such as the Discrete Element Method [e.g. Luding, 2008; Chen, 2009; Jajcevic *et al.*, 2013; Kloss *et al.*, 2012; Pietsch *et al.*, 2018]. In addition to direct numerical simulation, process-based erosion models such as the HR model [Hairsine and Rose,

Chapter 1. Introduction

1991, 1992a, b] and landscape evolution models [Willgoose, 1989; Howard, 1994; Theodoratos *et al.*, 2018] represent the raindrop impact via empirical relations or as one of the factors in a more general formula. In order to develop and improve these different physical models, laboratory work is indispensable to obtain more detailed observations and find unknown parameters and interaction properties. In the last two decades, thanks to the development of high-speed camera technology, it is feasible to capture grain-scale processes with high resolution. In Table 1.2, some works in this area are reviewed.

Table 1.2 – Some laboratory works on single drop impact on granular beds.

Author(s)	Remarks
Royer <i>et al.</i> [2005]	They studied the granular jet formation of highly loose dry fine particles by dropping a falling solid sphere. They showed the narrowing effect in the splashed dry particles even when there was no surface tension. This study highlighted the significance of dynamic coupling between the granular material and gas flow.
Furbish <i>et al.</i> [2007]	Rain splash of dry sand was revealed at different surface slopes using a high-speed imaging and sticky paper splash targets. They found a similar ejection distance for different droplet sizes. However, the number of detached particles increased by larger droplet sizes.
Katsuragi [2010]	Considering different droplet impact speeds and grain sizes, he found a power law ($\frac{R}{R_w} = \frac{\rho_g}{\rho_w} We^{\frac{1}{4}}$) for the resulted crater radius (R) as a function of the Weber number ($We = \frac{2\rho_w R_w v^2}{\lambda}$) where R_w , ρ_w , ρ_g , v and λ are the droplet radius, water density, grain bulk density, impact velocity and surface tension, respectively.

Continued on next page

Table 1.2 – continued from previous page

Author(s)	Remarks
<i>Marston et al.</i> [2010]	Six types of liquid were dropped into the powder with various packings. They found that the Weber number and packing fraction are the main parameters in determining the resulted surface shape after the droplet impact.
<i>Katsuragi</i> [2011]	He studied the impact depth and cratering radius. Variety of droplets size, liquid types, sediment bed materials and particle sizes were examined. He found that the time scale of penetration was proportional to the square root of the droplet viscosity.
<i>Emady et al.</i> [2011]	They subjected three physically different powders to three types of liquid drops. Tunneling occurred on the surface of the cohesive powder beds. However, for coarser powders, granules were formed by a spreading mechanism at a low impact velocity.
<i>Delon et al.</i> [2011]	The crater formation was studied by considering droplet size and velocity, as well as the grain size. The maximum (D_{max}) and final (D_{final}) diameters were compared as a function of the Weber number. There was a critical the Weber number which rendered the maximum ratio of D_{max}/D_{final} .
<i>Hamlett et al.</i> [2013]	Hydrophobic and hydrophilic granular materials were subjected to droplet impacts. The hydrophobic particles showed three phases of rebound, pinning and fragmentation whereas the hydrophilic particles revealed solely the pinning state.

Continued on next page

Table 1.2 – continued from previous page

Author(s)	Remarks
<i>Long et al.</i> [2014]	They used a three-dimensional, time-resolved, particle tracking approach to investigate the impact of a liquid droplet onto a granular bed. It was found that only about 0.2 % of the impact momentum was transferred into the ejected particles.
<i>Zhao et al.</i> [2015]	They measured the spreading diameter of droplets, their penetration into the surface and morphology deformation. Different impact velocities and granular materials with various packing fractions were tested (droplet size = 2.8 mm, particle size = 70-110 μm , $\rho_s = 2500 \text{ kg s}^{-1}$). It was pointed out that the grains underneath the droplet splash had a critical packing fraction of $\phi^* = 0.588$ during the impact regardless of the initial packing fraction.
<i>Zhang et al.</i> [2015]	The scale of craters generated on wet granular media was explored (Droplet size = 1.8 mm-5.3 mm, particle size = 90 μm). They used mineral oil to neglect the evaporation effects to the extent possible. Moreover, they tracked the deformation of the surface using a scanner and measured compressive stresses at different saturation rates (0-0.8 %). They showed that the unique energy partition originally proposed for liquid-drop impact cratering in dry granular media also applied for wet media.
<i>Joung and Buie</i> [2015]	They found that aerosol can be generated by raindrop impact. A high-resolution camera captured the bubbles by which the elements of the porous media delivered into the environment.

Continued on next page

Table 1.2 – continued from previous page

Author(s)	Remarks
<i>Ryżak et al.</i> [2015]	Surface images after one, five, and ten drop impacts were recorded to find out differences between the consecutive impacts. They observed that the deformed surface area and the number of detached droplets increased by repetition of the impacts.
<i>Katsuragi</i> [2016]	This book includes the theories behind soft impacts and cratering phenomena.
<i>Lee et al.</i> [2016a]	Impacts on three different natural porous stones were investigated at various drop velocities. It was shown that apart from low impact velocities, the spreading diameter increased by increasing the Weber number.
<i>Lee et al.</i> [2016b]	A numerical simulation (CFD-VOF) was validated against the experimental data for the impact of three different liquids on porous media. The validation was used to find the principal model input parameters. They reported that viscous dissipation could not be neglected at low impact velocities.
<i>Lardier et al.</i> [2018]	They studied the drop impact on two beds of pure sand and loamy soil. It was found that the spread of drop was more anisotropic on the loamy sand due to the wider distribution of particle sizes.

1.4.2.2 Unchanneled surfaces

In this section, we review some laboratory works on sediment transport at hillslopes where no rill is formed on morphology (Table 1.3). Unchanneled surfaces can be generated in two ways. First, when the shear stress of the overland flow is less than the critical shear stress needed to detach particles from the surface and make a

Chapter 1. Introduction

rill. Second, when the shear stress is sufficient to detach particles, but the droplet momentum is adequately high to smooth morphology. In other words, the advective (rilling) effect of the overland flow is less than the diffusive (smoothing) effect of droplet impacts [Sweeney *et al.*, 2015].

Table 1.3 – Laboratory-scale studies of rainfall driven sediment transport on unchanneled surfaces.

Author(s)	Remarks
<i>Neal</i> [1938]	On a 1.1-m×3.66-m flume, he studied the effect of slope and rainfall intensity on erosion and runoff. It was shown that on an initially dry surface, runoff did not occur until several minutes and after a certain time, it reached a fixed value. Moreover, he found that erosion intensified by increasing slope and rainfall intensity.
<i>Heilig et al.</i> [2001]	They designed a simple experiment on a 1.1-cm×3.66-cm soil surface to validate the HR model. The results clearly revealed the formation of a shield layer, and the quantitative data were in a good agreement with the analytical solutions of the HR model.
<i>Gao et al.</i> [2003]	On a circular (diameter = 7 cm) soil surface, the effect of ponding depth and soil detachability was investigated on soil erosion. Detachability was a constant value up to a critical overland flow depth and considerably decreased for larger depths.

Continued on next page

Table 1.3 – continued from previous page

Author(s)	Remarks
<i>Gao et al.</i> [2005]	By the same experimental setup of <i>Gao et al.</i> [2003], they saturated the soil with chloride and applied different rainfall intensities to measure chloride concentration change over time. By including splash erosion effects in the transport equation of the solution, they developed a model for chloride transport that was in agreement with the experimental data.
<i>Walker et al.</i> [2007]	In an experiment similar to that of <i>Gao et al.</i> [2003], they concluded that infiltration reduces the soil erosion rate. They also showed that the HR model can predict this behavior.
<i>Tromp-van Meerveld et al.</i> [2008]	Using flume experiments (6-m×2-m), they tested the HR model by using measured and modified settling velocities. This study showed that the optimized settling velocities for finer and coarser particles were higher and lower than calculated from Stoke's law, respectively.
<i>Jomaa et al.</i> [2010]	They validated a 1-D HR model at different flume widths (with 6-m length) and showed that larger particles had a more scattered erosion than the finer particles.
<i>Armstrong et al.</i> [2012]	Formation of a shield layer on the soil was tested by considering the water depth, infiltration and slope of a 25-cm×25-cm soil surface. They observed that shield layer generation intensified by increasing the ponding water depth, slope and infiltration. However, the effect of infiltration was lower than the other factors.

Continued on next page

Table 1.3 – continued from previous page

Author(s)	Remarks
<i>Jomaa et al.</i> [2012a]	They measured soil erosion rate in the presence and absence of stone arrangement on a 6-m×2-m flume surface. It was shown that the erosion rate was proportional to the exposed soil.
<i>Jomaa et al.</i> [2012b]	Capability of the 1-D HR model was tested when there are rock fragments on the soil surface. They concluded that with high stone coverages and low rainfall intensities, the HR model is less accurate.
<i>Jomaa et al.</i> [2013]	Soil erosion rates were investigated in the presence of rock fragments and multiple rainfall events. The results revealed that for a short period of time, erosion is dominated by initial condition and afterward, it is mostly controlled by rainfall intensity.
<i>Dong et al.</i> [2013]	They introduced a power function to calculate the concentration of dissolved potassium in the soil and compared the model with an experiment (on a 1-m×0.4-m soil surface). They concluded that the solute concentration was affected by the initial soil moisture, rainfall intensity and slope.
<i>Sadeghi et al.</i> [2017a]	At different precipitation rates and flume slopes (6-m×1-m), they measured the soil particle distribution during the splash process by inserting local cups on the flume surface. They pointed out that the particle distributions at both upward and downward of the flume-slope were different from the original soil.

Continued on next page

Table 1.3 – continued from previous page

Author(s)	Remarks
<i>Hu et al.</i> [2018]	On a circular soil surface (diameter = 10 cm), they separated the effect of initial internal forces of particles and splash effects. The results revealed that the soil's initial forces could have more than 65% contribution to the splash erosion.
<i>Kiani-Harchegani et al.</i> [2018]	On a 6-m×1-m flume, they compared the grain size distribution of suspended sediment with original soil at different rainfall intensities and slopes. It was found that the proportion of finer particles in the eroded sediment was more than that of the original soil.

1.4.2.3 Channeled surfaces

In this category of laboratory-scale investigations, advective processes (shear-driven erosion) overcome the diffusive ones so that rills are generated on the surface. The experiments include physical interpretation and description (initiation of rills, knickpoint and headcut migration, minimization of energy expenditure, etc.) and/or geometrical characterization (drainage density, scaling laws, fractal features, etc.) of rill networks. The selected studies are presented in Table 1.4.

Table 1.4 – Experimental investigations of sediment transport and network analysis on incised beds.

Author(s)	Remarks
<i>Flint</i> [1973]	On a 45-cm×29-cm box, he found an exponential growth of the network and its drainage density. He carried out two experiments with maximum reliefs of 16 cm and 13 cm where at the lower relief, a more dendritic network was reported.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Mosley</i> [1974]	He studied on soil erosion at 90 mm h^{-1} rainfall intensity and different slopes (3-11 %) on a 15.3-m×9.2-m flume. Three initial surfaces of plane, divergent and convergent were considered. By using these experiments, six different types of erosion (sheet flow, micro rilling, rain splash, rill incision, recession of rill headcuts and lateral smoothing of rills) were qualitatively described.
<i>Parker</i> [1977]	On a 15.3-m×9.2-m flume, he analyzed rill formation on an artificial non-smooth surface with the intersection of two plates. He found that at low-reliefs, the network had a “headward growth” where the internal variation within the network was negligible. On the other hand, he identified considerable rearrangement within the incised network for the high-relief scenario which he named “Hortonian growth”.
<i>Phillips and Schumm</i> [1987]	They changed the main-slope of a 3-m×2-m flume from 1.1 to 16% and described rill networks’ structures. The experiment revealed that by increasing slope, the network becomes less dendritic and the junction angles to the mainstream decreases.
<i>Bryan and Poesen</i> [1989]	The effect of slope length on runoff, infiltration and rill development was studied. To this end, they divided a 17-m×0.85-m flume into seven sections and observed that the infiltration pattern was heterogeneous in the space because of surface sealing and incision. Moreover, they concluded that it is not possible to have a generally applicable relationship between the slope length and discharge.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Wittmann et al.</i> [1991]	They generated a dendritic area using a rotating nozzle and found that the rill network's fractal dimension linearly increased by increasing the water flow rate.
<i>Slattery and Bryan</i> [1992]	They studied erosion and rill formation on a 15-m long flume and 8.7% slope. It was revealed that the rills developed via the establishment of a knickpoint after a supercritical flow. However, they could not clearly define the rill initiation by a hydraulic metric.
<i>Gomez and Mullen</i> [1992]	An incised surface was generated on a 6.1-m×1.8-m flume and they identified the initiation, extension and abstraction of the networks that were explained by <i>Glock</i> [1931].
<i>Czirók et al.</i> [1993]	On a 60 cm×18 cm surface, the self-affinity of roughness was studied, resulting in the temporal and spatial scaling exponents of 0.9 and 0.78, respectively.
<i>Crave et al.</i> [2000]	An experiment was performed on a 14-cm×20-cm box where the droplet size was about 5 μm so that raindrop impacts did not disturb the fluvial field. It was observed that the average elevation exponentially decreased during about 160 min of rainfall. Moreover, a power law with an exponent of -0.045 was reported for the relation between drainage area and slope.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Hasbargen and Paola</i> [2000]	After generating a thoroughly dissected domain (a circular area with a diameter of about 1 m) with a rainfall intensity and an uplift rate of 6.5 and 2.8 $\mu\text{m s}^{-1}$, respectively, it was found that the ridges were regularly migrating and reordering during the experiment. Moreover, they measured an oscillating erosion rate around the uplift rate and related the oscillation to knickpoint migration in the domain.
<i>Brunton and Bryan</i> [2000]	The effect of network structure on sediment and water fluxes was studied on a 7.1-m \times 2.4-m flume. They generated three different networks and related the sediment flux variation to morphological processes (bank collapses, tributary development, knickpoint incision). This study highlighted the necessity of recording the rill network characteristics to have a more precise measurement and interpretation of water and sediment fluxes.
<i>Hancock and Willgoose</i> [2001]	An experimental landscape evolution on a 1.5-m \times 1.5-m flume was compared with a physically-based model (SIBERIA) results. For comparison, they used the hypsometric curve (the normalized drainage area versus normalized elevation), width function, cumulative area distribution, and area-slope relationship. It was concluded that SIBERIA could correctly simulate the experimental model landscape.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Römkens et al.</i> [2002]	They studied erosion and rill formation on rough, medium and smooth surfaces with temporally decreasing and increasing rainfall intensities (flume size: 3.75 m×0.6 m). It was shown that the erosion rate was higher for the rough surfaces. Furthermore, surface topography strongly affected the runoff distribution and the rill network structure.
<i>Hasbargen and Paola</i> [2003]	They showed that the spatial and temporal evolution of landscapes is intrinsically stochastic, even at constant external forcing factors (mainly uplift and rainfall) of the system. The sediment box was the same as that used in <i>Hasbargen and Paola</i> [2000]. They also found an exponent in the range of 0.13-0.16 for the slope-area power-law relations.
<i>Pelletier</i> [2003]	On a 15-m×9.2-m flume, they pointed out that the initial main-slope patterns have a substantial effect on morphology evolution and also on the final configuration of a rill network.
<i>Lague et al.</i> [2003]	With an experimental setup similar to that of <i>Crave et al.</i> [2000], they investigated the effect of initial morphology on the transient and steady-state conditions. They found that the initial morphology plays a significant role in the transient phase of evolution. Furthermore, a power law relation with an exponent of -0.12 was resulted for slope-drainage area relation, regardless of the initial condition and uplift rate.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Bonnet and Crave</i> [2003]	With the same experimental setup of <i>Crave et al.</i> [2000], they studied the effect of climate variation on the mean elevation of a basin. They observed that at a constant uplift rate, by a sudden increase of the rainfall rate, the average elevation decreased. However, a sudden decrease of the rainfall intensity induced a climatic uplift. They discovered that the fluvial network can be affected by climate forcing in the same range of tectonics.
<i>Gómez et al.</i> [2003]	Using a 4-m×2-m flume, rill network evolutions at two slopes (5 and 20 %) and different initial surfaces (low, medium and high roughness) were investigated. They concluded that incised surfaces (at 20 % slope) evolved according to the principle of minimum energy dissipation formulated for river networks. However, when splash played the dominant role in the erosion process (at 5 % slope), the energy dissipation calculated by the catchment-scale formula did not decay during the surface evolution.
<i>Raff et al.</i> [2004]	They studied the evolution of an incised surface on a 10-m×3-m flume at two different slopes (9° and 5°). They observed that by increasing the slope, the average depth and width of the rills increased while the ratio of width to depth decreased.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Rieke-Zapp and Nearing</i> [2005]	They considered five slope shapes under a constant rainfall intensity of 60 mm h^{-1} and studied rill formation by describing the patterns after 90 mins of rainfall (flume size: $4\text{-m}\times 4\text{-m}$). It was reported that the energy expenditure was a physical measure for self-organization of the rill networks. Moreover, they observed the measured drainage density to be relatively independent of the slope shape.
<i>Babault et al.</i> [2005]	The experiment consisted of a $40\text{-cm}\times 60\text{-cm}$ sediment box, surrounded by a plateau area. The central box ($40\text{-cm}\times 60\text{-cm}$) was raised by a constant uplift rate while eroded sediments were deposited on the plateau. The average elevation and denudation rates were measured. It was pointed out that when the time scale of deposition was higher than the ones for relief development (the central part), the relief entered a dynamic equilibrium. The dynamic equilibrium was defined as when the denudation equaled the relative uplift rate (the tectonic uplift - the rate of deposition). This study highlighted the contribution of piedmont sedimentation to the high elevation of mountain peaks.
<i>Bigi et al.</i> [2006]	On a circular area with a diameter of 1 m, they applied a uniform rainfall distribution of $10.8 \mu\text{m s}^{-1}$ (droplet size $< 200 \mu\text{m}$) and an uplift rate of $3.1 \mu\text{m s}^{-1}$ for 13 h. They observed the effectiveness of knickpoints along streams on hillslope failures. It was concluded that the landslides were more probable to occur at the downstream of knickpoints.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Turowski et al.</i> [2006]	They studied the effect of uplift rate on rill networks' geometries on a 40-cm×60-cm sediment box with different rainfall (45 and 140 mm h ⁻¹) and uplift (0.1-2.5 cm h ⁻¹) rates. They observed that the channels' width, depth, and cross-sectional area increased with discharge as a power law. Moreover, channel slopes increased linearly with increasing the uplift rate and were independent of discharge.
<i>Babault et al.</i> [2007]	In a similar experiment as that of <i>Babault et al.</i> [2005], they showed the piedmont growth into the upstream and smoothing of the incised morphology even in the absence of diffusive effects.
<i>Yao et al.</i> [2008]	Rill formation on a cohesive sediment was investigated on an 8-m×3-m flume. They measured rill initiation, flow depth and critical shear stress at different rainfall intensities and slopes. They found that rill initiation was more affected by the flume slope than the rainfall intensity.
<i>Tatard et al.</i> [2008]	On a 10-m×4-m flume with 1% slope, they measured the velocity field using the Scalable Vector Graphics technology. They reported that the friction factor decreased by increasing the Reynolds number ($Re = \frac{4ur}{\nu}$ where u , r and ν are average velocity, hydraulic radius and fluid kinematic viscosity, respectively). In addition, they reported less sensitivity of the friction factor to soil roughness at higher Re .

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Berger et al.</i> [2010]	Using a 2-m×1-m box, they considered three different slopes (10, 20 and 30 %) and rainfall intensities (60, 90, 120 mm h ⁻¹) and analyzed rill development. Compared to the slope, rainfall intensity had a larger effect on the rill formation. Furthermore, they reported that the theory of minimization of energy expenditure was capable of describing the rill network establishments.
<i>Gordon et al.</i> [2011, 2012]	They conducted experiments on a 7.0-m×2.4-m flume and non-cohesive sediment. They studied the response of a rill network to two immediate reductions of the base level. They found that significant soil erosion occurred when the base level was dropped and headcuts actively migrated upstream. However, a few minutes after the base level reduction, the sediment fluxes decreased and the rill network migration stopped.
<i>Shit et al.</i> [2013]	They carried out five experiments with slopes of 15°, 20° and 25°, and rainfall intensities of 60, 90 and 120 mm h ⁻¹ . Rill networks were visualized and the rill sizes were reported. They found that interrill erosion decreased as the rills developed.

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>McGuire et al.</i> [2013]	They used a numerical model [<i>Simpson and Castellort</i> , 2006] in which the Digital Elevation Model of the flume was used as the initial condition. The numerical model calculated the direct material transport due to the rain splash, fluvial transport and sediment deposition. The deposition term in the model did not have a significant effect on the model-experiment agreement. They also showed that the direct impact of raindrops was an essential factor in limiting the entrainment rate and that the rill network geometries were dependent on the relative strength of advective and diffusive processes.
<i>He et al.</i> [2014]	Two types of soil (clay loam and loess soils) were subjected to two rainfall intensities on a 5-m×1-m flume at 10 % slope. They pointed out that the erosion rate of the loess soil was intensified as the rill was formed on the surface whereas the loam soil had the same sediment flux before and after the rill generation (due to the surface cohesion).
<i>Bennett et al.</i> [2015]	On a 7.0-m×2.4-m flume, they described surface incision of a cohesive soil as a result of an episodic base-level lowering. The bifurcation, length ratios and fractal dimension had more oscillation at the beginning of the experiment and reached a steady value after a relatively short time.
<i>Bennett and Liu</i> [2015]	They assessed the self-similar characteristics of an incised surface of cohesive sediment on a 7.0-m×2.4-m flume and found an exponent of 0.49 for the Hack's law [<i>Hack</i> , 1957].

Continued on next page

Table 1.4 – continued from previous page

Author(s)	Remarks
<i>Shen et al.</i> [2015]	They generated rill networks on a cohesive soil surface at two rainfall intensities of 50 and 100 mm h ⁻¹ (10-m×3-m flume). They measured soil loss and different geometrical characteristics such as mean rill width, depth and inclination angle, rill density, and rill tortuosity complexity. Overall, a more complex network was reported at the lower rainfall intensity.

1.5 Research questions and methodologies

As described in Table 1.3, previous works on unchanneled surfaces concentrated more on erosion rate than on the morphological changes. Generation of a shield layer due to deposition of larger particles is mentioned ubiquitously in the literature. However, it is not clear if the slight particle rearrangements (without rill formation) can lead to hysteretic sediment fluxes for rainfall-driven erosion. Furthermore, the patterns that the deposited particles make on the surface (micro-roughness) are unknown. Therefore, introducing some metrics for unchanneled morphology evolutions enhances future modeling and experiments by quantifying the expected micro-roughness condition during the process. Apart from a geometrical quantification, having more physical insights about the unchanneled surface evolutions is crucial. In the flume-scale studies, the calibrations of physically-based morphology evolution models are limited to the steady-state condition of incised surfaces, and the transient states of the models are not compared with the measurements. Additionally, it is not known how efficiently a large-scale LEM can capture the dynamic changes of morphologies when there is no channeling. In this dissertation, the objective is to investigate the details of high-resolution overland flow morphologies at the laboratory scale. To this end, the statistical and physical aspects of the morphology evolution are considered as below:

D) The effect of subtle morphological changes on the hysteresis patterns of rainfall-driven sediment fluxes:

Hysteresis loops are omnipresent for shear-driven sediment transport in river flows [e.g., *Klein*, 1984; *Williams*, 1989; *De Girolamo et al.*, 2015; *Sun et al.*, 2016; *Dean*

Chapter 1. Introduction

et al., 2016; *Sherriff et al.*, 2016]. In this work, we investigate the same patterns for rainfall-driven sediment transport. The HR model (section 1.3.1) along with a flume experiment is implemented to study the effect of morphological change (generation of a shield layer as a result of particle deposition) on the hysteresis loops of different sediment particle sizes (Chapter 2). Erosion rates are measured during a temporally variable precipitation rate. The HR model provides the theoretical support for the descriptions by calculating the generation of the shield layer on the soil surface which is extremely difficult to measure on the flume.

II) Statistical characterization of overland flow morphologies:

After highlighting the importance of micro-roughness variations in the dynamics of the erosion phenomenon, the geometrical structure of unchanneled surfaces is statistically investigated under a spatially non-uniform rainfall (Chapter 3). The details of morphology are mainly affected by deposited particles. In the analysis, the non-incised surface is characterized as a network analogous to large scale river networks (section 1.4.1). Based on the extracted network, the self-similarity of the system is presented as power laws of exceedance probabilities of discharge and drainage area, and upstream length.

III) Applicability of a physically-based catchment-scale model to unchanneled morphologies at the laboratory scale :

After statistically characterizing the unchanneled morphology evolutions, in Chapter 4, a physically-based LEM (section 1.3.3) is tested at the flume-scale. Although there is no observable rill on the surface, the continuous flow is represented by a discharge network. The LEM is numerically solved and calibrated to simulate the experimental data. The objective is to find out the strong and weak points of a typical catchment-scale model for a diffusive dominated domain at the flume-scale.

In order to resolve more details of morphology via the LEM, some modified approaches are tested. Based on the roughness probability distribution function, the LEM is applied as a stochastic partial differential equation in different scenarios. The model is modified by considering heterogeneous parameters in seven different scenarios. The modified models are calibrated to reproduce the scaling laws found experimentally in Chapter 3. Then, the simulation results are critically compared with the experimental and the original model to suggest the best modification approaches.

2 Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

Mohsen Cheraghi¹, Seifeddine Jomaa², Graham C. Sander³ and D. A. Barry¹

¹Ecological Engineering Laboratory (ECOL), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

²Department of Aquatic Ecosystem Analysis and Management, Helmholtz Center for Environmental Research—UFZ, Magdeburg, Germany

³Department of Civil and Building Engineering, Loughborough University, Loughborough, UK

Published in Water Resources Research:

Cheraghi, M., Jomaa, S., Sander, G. C. and Barry, D. A. (2016). Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects. *Water Resources Research*, 52(11), 8613–8629. <https://doi.org/10.1002/2016WR019314>.

2.1 Introduction

Estimates of temporal variations of suspended load with discharge are needed for the assessment of aquatic ecosystems, estimates of contaminant export from catchments, and the prediction of stream water quality [Walling and Webb, 1985; Wood and Armitage, 1997; Batalla *et al.*, 2004; O'Connell and Siafarikas, 2010; Rossi *et al.*, 2013; Halliday *et al.*, 2014; Karimae Tabarestani and Zarrati, 2015; Lloyd *et al.*, 2016; Rose *et al.*, 2018]. The relationship between discharge and sediment concentration is available for different catchments and rivers [Klein, 1984; Williams, 1989; Seeger *et al.*, 2004; Nadal-Romero *et al.*, 2008; Sadeghi *et al.*, 2008; Smith and Dragovich, 2009; Eder *et al.*, 2010; Alemayehu *et al.*, 2014; De Girolamo *et al.*, 2015; Sun *et al.*, 2016; Dean *et al.*, 2016; Sherrieff *et al.*, 2016]. Depending on the variation of discharge and sediment concentration versus time, different hysteretic concentration-discharge curves can be generated including clockwise, anticlockwise, and figure eight. At the plot scale, Strohmeier *et al.* [2016] examined soil erosion under temporally variable rainfall at two different locations and assessed the effect of extreme rainfall events on long-time erosion. They reported that a few extreme rainfall events can have a permanent effect on land degradation in contrast to lower intensity but frequent events. Their results also show the opposite case, i.e., that the soil loss can be mainly caused by a large number of low intensity rainfall events rather than the rainfall extremes. The different patterns were due to spatially variable factors in the landscape such as soil type, land use, and slope.

Hysteresis loops are a feature of plot-scale and catchment-scale sediment transport. Several studies investigated the factors and processes responsible for these loops in order to interpret or determine the distribution of sediment sources within a catchment [Seeger *et al.*, 2004; Smith and Dragovich, 2009; Yeshaneh *et al.*, 2014; Hamshaw *et al.*, 2018]. The difficulty of interpretation at these scales is that there are complications arising from spatial and temporal variability in climate [Ghahramani and Ishikawa, 2013; Arjmand Sajjadi and Mahmoodabadi, 2015; Dai *et al.*, 2016], soil types [Keesstra *et al.*, 2014; Rodrigo Comino *et al.*, 2016], land use [Cerdà *et al.*, 2009; Prosdociimi *et al.*, 2016], topography [Ghahramani *et al.*, 2012], catchment connectivity [Ghahramani and Ishikawa, 2013; Marchamalo *et al.*, 2015; Masselink *et al.*, 2016], channel storage and bank erosion [Buendia *et al.*, 2016], and soil saturation and the initial condition of the surface soil [Seeger *et al.*, 2004; Bussi *et al.*, 2014; Kim and Ivanov, 2014; Pietroń *et al.*, 2015]. Hysteresis patterns are generally seen as complex

and their interpretation is not straightforward [Gao and Josefson, 2012; Aich *et al.*, 2014; Juez *et al.*, 2018].

Simplified laboratory systems are more amenable to develop understanding of specific processes. For instance, hysteresis loops can be obtained under a single-peak individual storm event on a planar landscape. For this case, Sander *et al.* [2011] conducted simulations using the Hairsine and Rose (HR) model [Hairsine and Rose, 1991, 1992a, b] in which variations in the initial condition of the deposited layer (i.e., previously eroded soil) were imposed. They found that not only were all the specific forms of the hysteretic loops of Williams [1989] (i.e., clockwise—using a well-developed spatially uniform deposited layer at t (time) = 0, counter-clockwise—using no deposited layer at $t = 0$, and figure eight—having a spatially varying deposited layer at $t = 0$) straightforward to reproduce, but they could also replicate the same patterns found in catchment studies by Eder *et al.* [2010] and Oeurng *et al.* [2011]. Subsequently, Zhong [2013] extended the results of Sander *et al.* [2011] to demonstrate that a variety of multilooped hysteresis patterns could be obtained without the need for flows over a complex topography or for multiple storm events.

The work of Sander *et al.* [2011] and Zhong [2013] shows that the HR model reproduces hysteretic loops in sediment concentration versus discharge as a result of two factors. First, the model accounts for the spatial variability in the distribution of easily erodible sediment at the start of an event. Second, it accounts for deposition as a separate rate process and as such directly models the preferential deposition of different sediment sizes resulting in the growth of a deposited layer having different erosive characteristics to the original soil bed. Thus, hysteresis is a result of interactions between the time-varying overland flow and differences between the cohesive strength of the original uneroded soil and deposited layer. Consequently “the [initial] spatial distribution and particle size composition of previously deposited sediment plays a significant role in determining the erosive response of the land surface” [Sander *et al.*, 2011].

The role of the surface soil composition and soil compaction due to rainfall was investigated by Jomaa *et al.* [2013], who reported experimental data and associated modeling (using the HR model) of rainfall-driven erosion. They applied multiple rainfall events separated by a drying period, which enabled them to investigate the effect of initial soil conditions (surface sealing, wetting-drying cycles, and initial moisture content). They found that “the soil erosion short-time response is mainly

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

controlled by the initial conditions, whereas the long-time behavior is controlled by the precipitation rate only.” Following this work, the importance of initial surface sediment conditions was considered by *Bussi et al.* [2014], who modeled sediment transport of the Goodwin Creek catchment. They found that the “estimation of the loose deposited sediments at the beginning of the storm event is fundamental for proper event scale modeling of soil erosion and sediment transport of the Goodwin Creek catchment”. *Bussi et al.* [2014] simulated the different hysteresis loops of *Williams* [1989], although they had mixed success in reproducing the correct orientation and loop size of the experimental data.

In more recent work on the role of initial conditions in soil erosion, *Kim and Ivanov* [2014] and *Kim et al.* [2016a, b] have recently further developed and confirmed the findings of *Sander et al.* [2011] and *Jomaa et al.* [2013] by carrying out a series of detailed numerical studies on the effect of the initial deposited layer in the HR model on erosion rates and transport of eroded sediment. *Kim and Ivanov* [2014] considered combinations of two consecutive 1 h storms of differing but constant intensities that were also separated by different time intervals. For combinations where the second storm had the same intensity, quite different sediment transport responses during the second storm were seen, as observed earlier by *Jomaa et al.* [2013]. They also found that this nonuniqueness in the erosive response was due to the first storm resulting in different compositions of the deposited layer (or initial conditions) prior to the start of the second storm. *Kim et al.* [2016a] also demonstrated the role of the subsurface initial moisture content in causing nonunique sediment transport under the same rainfall history. Numerical simulations reported by *Kim et al.* [2016b] on total sediment loss at the plot scale provide evidence (their supporting information Figure S2) of the dependence of clockwise and counter-clockwise hysteresis loops on the initial state of the deposited layer.

Previous studies at the field scale reported hysteresis loops for the total suspended concentration, but not the corresponding results for the different particle size classes [*Alemayehu et al.*, 2014; *De Girolamo et al.*, 2015; *Sun et al.*, 2016; *Dean et al.*, 2016; *Sherriff et al.*, 2016]. It is unknown whether all size classes have the same or different hysteretic behavior as the total concentration. At the laboratory scale, *Polyakov and Nearing* [2003] analyzed flow-driven erosion experiments in which they considered both steady state and time-dependent hysteretic conditions. The steady state experiments were carried out on an 8 m flume for two different inflow boundary conditions, either a zero or constant sediment flux, the latter being greater than the

transport capacity of the flow. The volumetric water discharge for both boundary conditions was the same. The transient experiments were performed on a 2 m flume for which the incoming sediment flux at the boundary was cycled every 0.25 h between the two fluxes used in the steady state experiment. Both sets of experiments displayed hysteresis in the total sediment concentration. Hysteresis in the suspended sediment concentrations for the individual size classes was also shown for the steady state experiment; however, size class data were not measured for the transient experiments. *Sander et al.* [2007] subsequently used the HR model to reproduce the steady state results of *Polyakov and Nearing* [2003] for both the total sediment concentration and for the different size classes. For the transient experiments, flow through the flume was supercritical and the bed morphology evolution was coupled with the overland flow. To account for this, *Sander et al.* [2011] extended the finite volume scheme of *Heng et al.* [2009, 2011], which combined the HR model with the Saint-Venant equations for flow and the Exner equation for modeling bed elevation changes. Figure 3 of *Sander et al.* [2011] shows that their model reproduces the rapid rise and fall in total sediment concentration at the end of the flume that results from the periodic sediment flux boundary condition along with the associated hysteretic behavior of the transported sediment.

There appears to be only one time-dependent laboratory study on hysteresis effects in overland flow sediment transport [*Polyakov and Nearing*, 2003], and it considers only the total suspended sediment concentration. There have been no experiments that specifically investigate the role of particle size in hysteretic transport other than the steady state data of *Polyakov and Nearing* [2003]. Consequently, the aim of this paper is to investigate hysteretic sediment transport under rainfall-driven erosion conditions using a well-controlled flume-scale experiment. The experiment involves a symmetric, single-peak rainfall event made up of seven sequential 20 min periods of differing constant intensities, with the peak intensity occurring for the fourth period. Throughout these seven periods, discharge and sediment size class data are measured at the flume outflow in order to quantify hysteresis in particle size class concentrations. The data are analyzed using the HR model, which is shown to reproduce the size class hysteretic behavior displayed by the experimental data. This investigation therefore compliments those of *Sander et al.* [2007, 2011] on flow-driven erosion to the case of rainfall-driven erosion.

2.2 Methods

2.2.1 Experiment

The study was carried out using the 6 m × 2 m EPFL soil erosion flume with two collectors at the outlet. Rainfall was applied to the lower 5 m of the flume's length from 10 oscillating valves that generate approximately uniform rainfall with a uniformity coefficient of 0.86 [Tromp-van Meerveld *et al.*, 2008; Jomaa *et al.*, 2010]. The precipitation rate is changed by varying the oscillation frequency of the sprinklers. Different parts of the flume are shown in Figure 2.1. More detailed descriptions of the flume characteristics are reported elsewhere [Viani, 1986; Baril, 1991; Jomaa *et al.*, 2010]. The flume was filled with an agricultural loamy soil with 4% clay, 29% silt, 41% sand, and 26% fine gravel from a field near Sullens in the Canton of Vaud, Switzerland. Soil characteristics are given by Baril [1991]. After ploughing and disaggregating the top-soil to a depth of 20 cm, a mechanical smoother was moved along the flume several times to ensure a uniform initial surface condition.

For loamy soils, the critical stream power above which entrainment occurs is in the range 0.15-0.20 W m⁻¹ [Beuselinck *et al.*, 2002]. In this work, the maximum stream power was estimated to be 0.013 W m⁻¹ and therefore raindrop-driven erosion was the dominant mechanism of sediment transport. No rills were observed during a visual post-experiment inspection. As seen in Figure 2.2, the 140 min precipitation period involved symmetric rising and falling limbs, divided into seven consecutive 20 min intervals, which are referred to as rainfall events and are denoted by E1 to E7, respectively. The rainfall intensity increased from 30 mm h⁻¹ (E1) up to 60 mm h⁻¹ (E4) on the rising limb. During the experiment, flume discharge was collected regularly in half-liter containers (Collectors 1 and 2, Figure 2.1). Because of higher erosion rate at the beginning of the experiment, sampling was performed continuously for the first 10 min and thereafter every 3 min. The size and proportion of the particle size classes are presented in Table 2.1. The collected samples were used to determine discharge rates and sediment concentrations of the total and individual size classes. Seven size classes were considered (Table 2.1), with concentrations denoted by C₁-C₇. For sediment concentration measurements from the collected samples, the larger size classes (>100 μm) were sieved, while for the rest a laser particle size analyzer was employed.

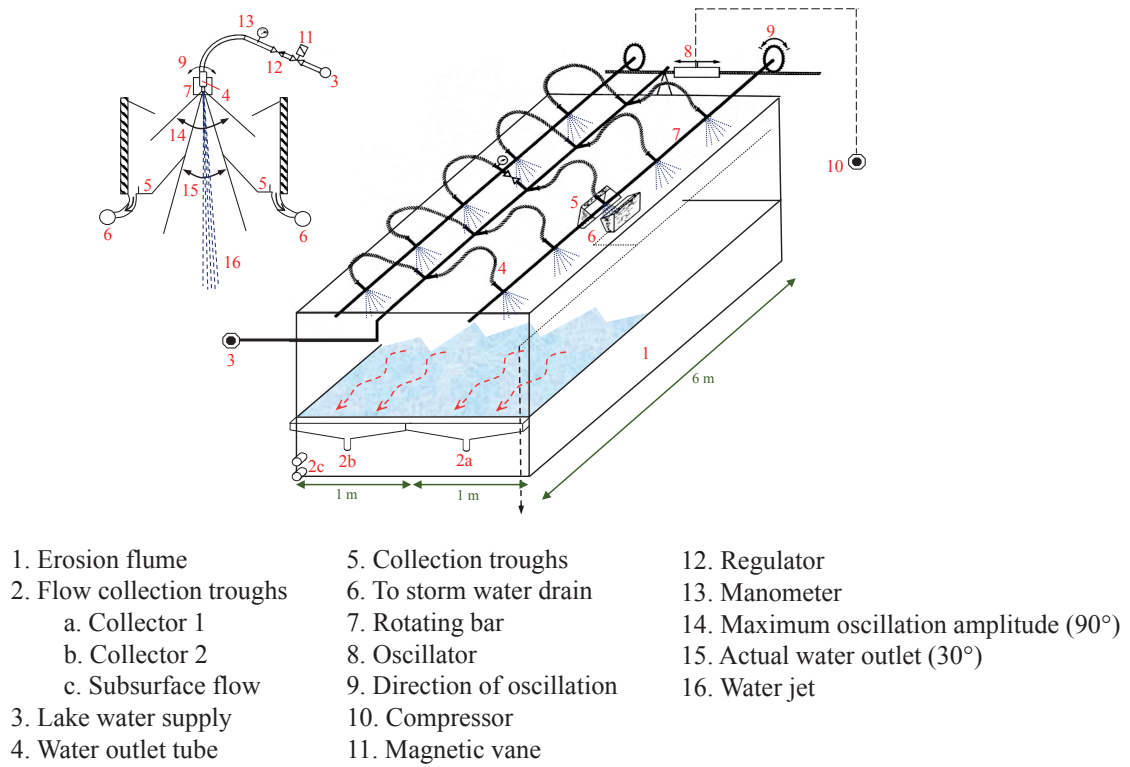


Figure 2.1 – Schematic of the EPFL soil erosion flume. The flume slope can be varied between 0 and 30%. Precipitation is applied using 10 oscillating sprinklers located 3-m above the soil surface.

Table 2.1 – Properties of seven different particle size classes of the soil. The settling velocities are from *Tromp-van Meerveld et al.* [2008].

Size Class	Diameter		Proportion (by Mass) in the Original Soil, p_i (%)	Settling Velocity, v_i (m s ⁻¹)		v_i (m s ⁻¹) Used in HR Model
	From	To		From	To	
C_1	0	2	3.7	8.0×10^{-8}	4.0×10^{-6}	5.0×10^{-7}
C_2	2	20	19.4	4.0×10^{-6}	4.0×10^{-4}	1.5×10^{-5}
C_3	20	50	8.3	4.0×10^{-4}	2.5×10^{-3}	7.0×10^{-4}
C_4	50	100	8.7	2.5×10^{-3}	1.4×10^{-2}	4.0×10^{-3}
C_5	100	315	17.7	1.4×10^{-2}	3.7×10^{-2}	4.0×10^{-3}
C_6	315	1000	20.6	3.7×10^{-2}	6.9×10^{-2}	4.0×10^{-3}
C_7	>1000		21.6	6.9×10^{-2}	1.4×10^{-1}	6.0×10^{-2}

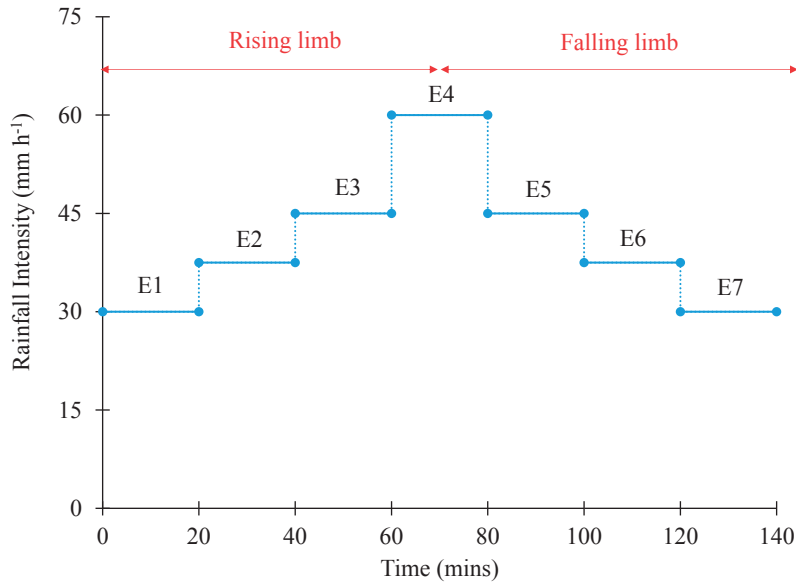


Figure 2.2 – Seven sequential rainfall events, each 20-mins long.

2.2.2 Model application and parameter estimation

The Hairsine and Rose model [Hairsine and Rose, 1991, 1992a, b] is described in section 1.3.1. An analytical approximation of the HR model was presented by Sander *et al.* [1996] and validated by different flume-scale and small-scale experiments [Heilig *et al.*, 2001; Tromp-van Meerveld *et al.*, 2008; Jomaa *et al.*, 2012b, 2013]. In their solution, a uniform suspended concentration is assumed within a constant depth of water, and spatial variability is ignored. The same approach was used in this study. With these assumptions and the combination of equations 1.1 and 1.2, the model consists of 14 coupled ordinary differential equations for seven particle size classes. These equations were solved for each 20 min rainfall event (E1–E7). Every 20 min, the final values of the sediment concentrations (C_i) and deposited layer masses (m_i) provided the initial conditions for the next rainfall event.

The settling velocity is an important parameter that determines the deposition rate of the individual particles. The ranges of settling velocity for different particle sizes were measured by Tromp-van Meerveld *et al.* [2008] who used the same flume and sediment (Table 2.1). For the larger particle size classes (C_4 – C_7 , Table 2.1), they used a 0.47 m tube filled with water and for the rest of the particles, the settling velocity was calculated using Stokes' law [Stokes, 1851]. They also fitted the model to the

experiments to find the optimal settling velocities, some of which were different from the measured values. As discussed in detail by *Tromp-van Meerveld et al.* [2008], possible explanations for this are flocculation, selective rainfall detachment, transport mechanisms rather than suspension, turbulence, hindered settling, the effect of infiltration, and measurement errors. The excess rainfall rate ($R = P - f$) was calculated based on the saturated infiltration rate (f) and the precipitation rate (P). In this experiment, the steady infiltration rates (f) for the different rainfall events were 4.82, 4.84, 4.23, 4.93, 4.24, 4.18, and 4.16 mm h⁻¹, respectively, for E1 to E7. Parameters fitted were detachability (a), redetachability (a_d), mass of the shield layer to protect the original soil (m^*), and water-layer depth (D). The calibrated values for the two collectors are presented in Table 2.2. An automatic calibration procedure was used for each rainfall event to deduce model parameters. The objective function was defined as the mean square error of the difference between model and experiment for the seven particle size classes, which was minimized using particle swarm optimization [Kennedy, 2010].

Table 2.2 – Optimized Parameters (a , a_d , m^* and D) for each rainfall event (E1-E7).

	Parameter	Rainfall Event						
		E1	E2	E3	E4	E5	E6	E7
Collector 1	a (mg cm ⁻³)	357	23	21	36	20	25	23
	a_d (mg cm ⁻³)	435	464	597	715	485	330	320
	m^* (mg cm ⁻²)	11.4	23.6	28.0	33.6	38.5	39.6	40.8
	D (mm)	6.1	7.1	8.0	10.5	8.4	8.3	7.9
Collector 1	a (mg cm ⁻³)	230	25	33	35	25	22	26
	a_d (mg cm ⁻³)	423	436	480	562	312	239	254
	m^* (mg cm ⁻²)	12.8	25.1	28.4	37.4	42.9	45.2	45.5
	D (mm)	4.8	5.4	5.7	6.7	6.0	5.6	5.4

2.3 Results and discussion

2.3.1 Experiment

2.3.1.1 Discharge

The discharge rates for each collector are shown in Figure 2.3. In the first rainfall event (E1, Figure 2.2), most rainfall infiltrated into the soil during the first 10 min. Afterward, the runoff increased up to a constant value as the soil became saturated, after which

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

infiltration into the soil was constant (4.82 mm h^{-1}). In the following rainfall events (E2–E7, Figure 2.2), runoff rates generally followed rainfall intensities (Figure 2.2 and 2.3). The higher discharge from Collector 2 compared to Collector 1 reflects the two-dimensional flow in the flume. Also, the discharge for the falling rainfall limb (E5–E7) is greater than that in the rising limb (E1–E4) for both collectors. This increase in discharge is due to reduced infiltration into the soil, caused by surface sealing and/or compaction of the soil resulting from raindrop impact. Additionally, due to compaction and overland flow, the surface roughness likely reduced over time, which resulted in increased surface flow rates.

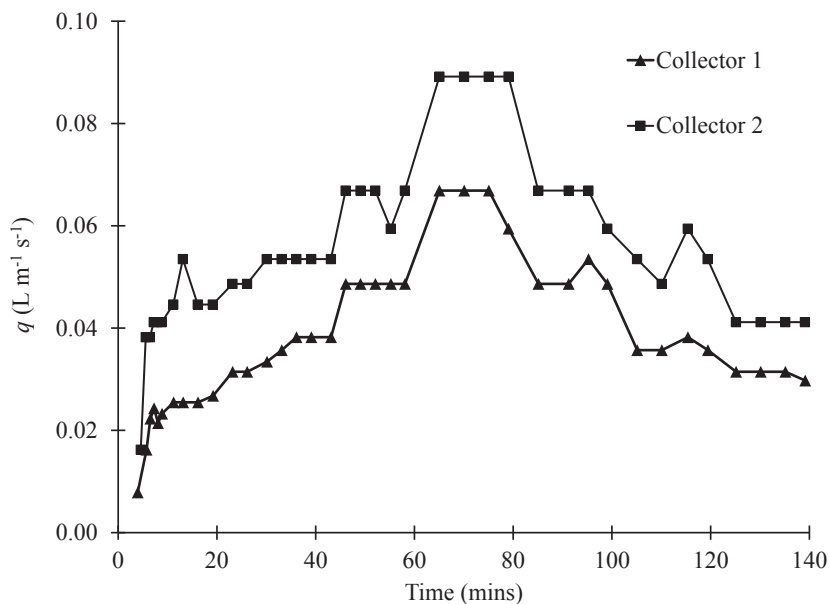


Figure 2.3 – Discharge measured at the two collectors. Except the first rainfall event (precipitation onto a dry soil), the discharge rapidly adapts to changes in rainfall intensity.

2.3.1.2 Sediment concentration

Figures 2.4 and 2.5 show the total sediment concentration (C) and the concentrations of individual particle classes (C_1 – C_7) for Collectors 1 and 2, respectively. The maximum total sediment concentration (C) occurs in the first rainfall event, in spite of the low precipitation rate in this period (30 mm h^{-1}). During each of the

subsequent rainfall events, C declines to a quasisteady equilibrium. The higher sediment concentrations in the first rainfall event (E1, Figure 2.2) stems from the initial condition—the soil was ploughed and not compacted prior to the experiment, leading to easily erodible soil. Additionally, there is the initial flush of fine material, which dominates the contribution to the early peak, and then the subsequent decline in C as the development of deposited layer reduces access to the finer soil particles [Sander *et al.*, 1996]. During rainfall events E2–E4 (rainfall intensities of 37.5, 45, and 60 mm h⁻¹, respectively), C shows a small increase at the beginning of each precipitation event and rapidly reaches a near-constant value. Finally, during the last three rainfall events (E5–E7), due to the decreasing rainfall intensity, C decreases. Observe that C is higher for the same precipitation rate during the rising limb compared to falling limb, i.e., C is lower during E7 than E1, E6 than E2, and E5 than E3. This is likely due to on-going compaction of the soil caused by raindrop impact and development of the shield layer, leading to reduced availability of erodible fine sediment. Both of these effects have been previously observed in flume experiments by Jomaa *et al.* [2012b] and field studies by Cerdà [2001], while Kim and Ivanov [2014] also noted the effect of the shield layer development on subsequent rainfall events with their 2-D numerical simulation at the large plot scale. We return to the shield layer below.

The results for the individual particle size classes (Figures 2.4 and 2.5) show that their temporal evolutions agree with, or are in contrast to, the overall behavior displayed by C . For instance, the maximum contribution of the two finest size classes (<2 and 2–20 mm, i.e., C_1 and C_2) is at the beginning of the first precipitation period E1, where C_1 reached 1.00 and 1.38 g L⁻¹ for Collectors 1 and 2, respectively, with the corresponding values for C_2 being 5.11 and 6.95 g L⁻¹. The calibrated parameters of the HR erosion model (Table 2.2) for this period also show that the soil detachabilities were greater than for the subsequent rainfall events, where the maximum concentrations decreased regardless of the rainfall intensity (except for a subtle increase occurring for 60 mm h⁻¹ during E4). After the second rainfall event, the proportion of the finer particles decreased in the deposited layer (see section 2.3.1.3) and hence also in the flume discharge. This trend was maintained during the increasing rainfall events of E3 and E4, indicating that the availability of finer sediments decreased over the course of the experiment.

Sediment concentrations were more sensitive to the precipitation rate variation for the middle particle size classes. For particle size C_3 , for both collectors (Figures 2.4

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

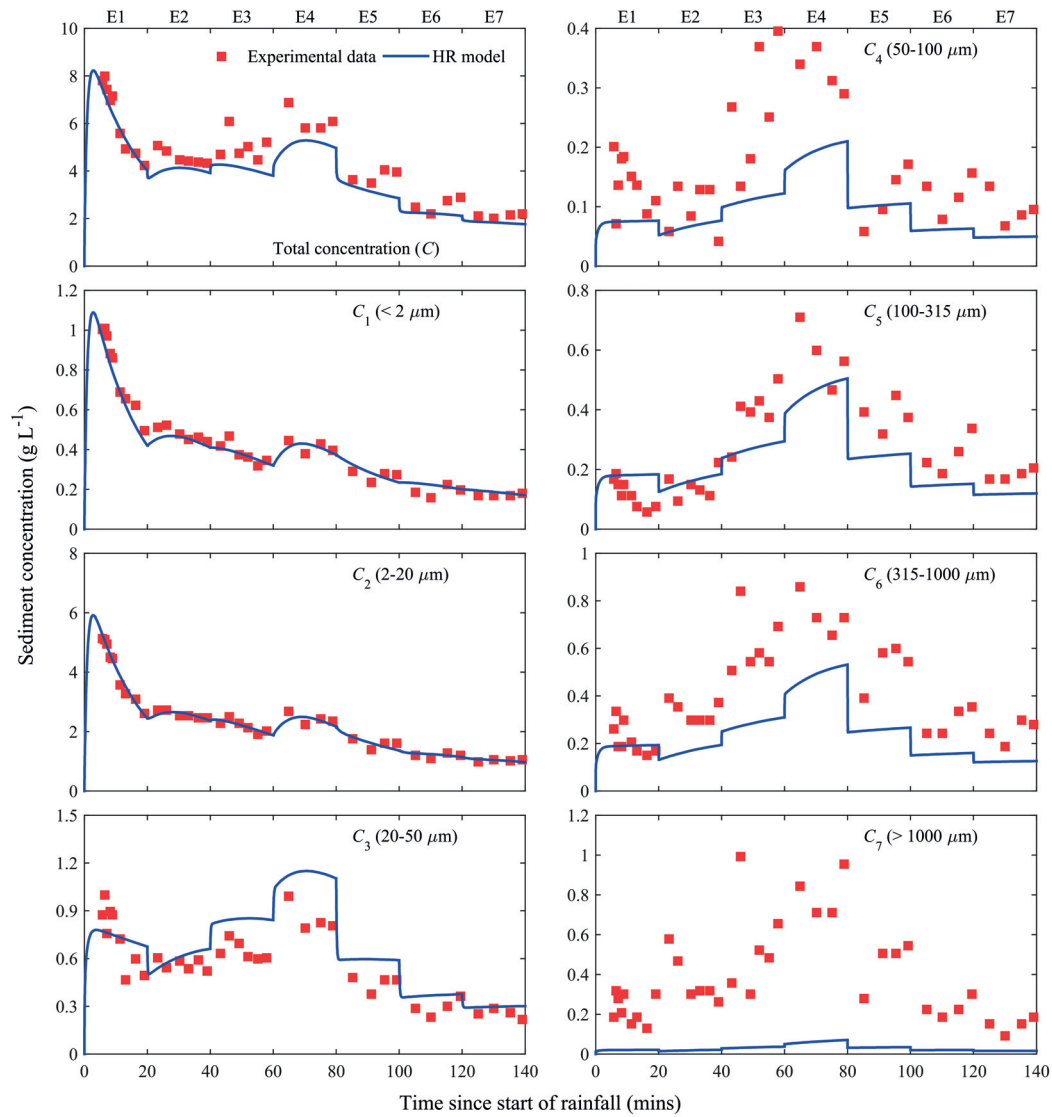


Figure 2.4 – Total sediment concentration (C) and the concentration of the seven different particle size classes ($C_1 - C_7$) for Collector 1. Regardless of the rainfall intensity, concentrations of the first two size classes decrease with time whereas concentrations of the larger particle sizes vary with the rainfall intensity.

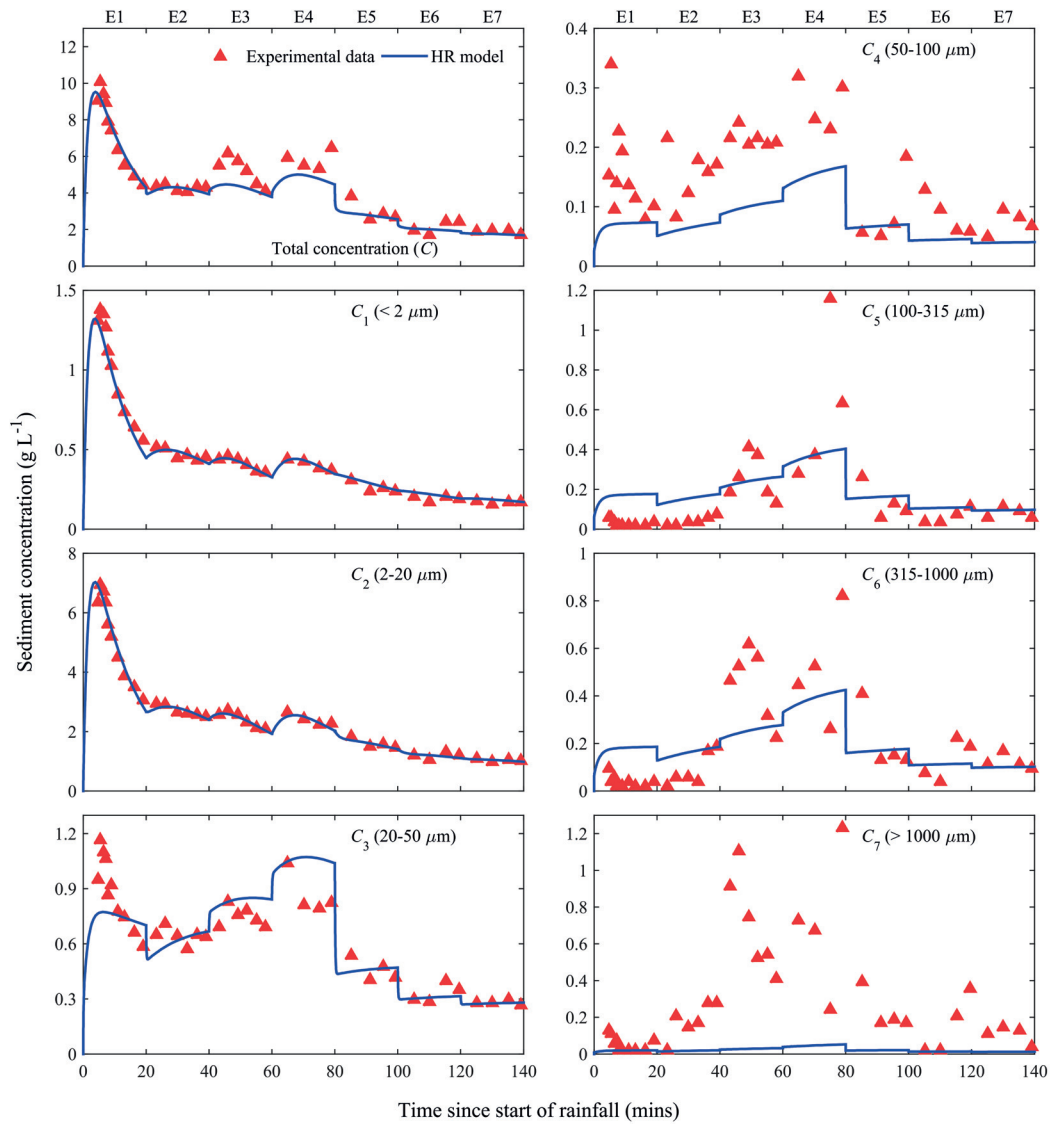


Figure 2.5 – Total sediment concentration (C) and concentrations of the seven different particle size classes ($C_1 - C_7$) for Collector 2. The HR model predicts the concentrations of the finer particle size classes satisfactorily. As the particles become larger, the difference between the model predictions and experimental data increases.

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

and 2.5), the concentration at the maximum rainfall intensity (60 mm h^{-1} , E4) was comparable to that of the first rainfall event (30 mm h^{-1} , E1). For particle size class C_4 , the two collectors both show that the greatest concentrations appear at the maximum rainfall intensity (60 mm h^{-1} , E4). For Collector 2 (Figure 2.5), the maximum concentration of C_4 (0.34 g L^{-1}) is at around 5 min, although this point is likely an outlier. The C_4 data from both Collectors 1 and 2 show that if this point were ignored, the same temporal behavior and the same magnitudes for each separate rainfall were measured. Data from Collectors 1 and 2 agree well for all of the size classes as well as total concentration, demonstrating consistent sediment transport on both sides of the flume.

The rainfall intensity had a noticeable impact on the transport of the coarser sediment classes (C_5 - C_7). At the precipitation rate of 30 mm h^{-1} , erosion rates of the large particles for the rising limb (E1) tend to be less than for the falling limb (E7), which is opposite to the behavior of the finer particle sizes. Although the condition of the topsoil changes between the second and sixth rainfall events (E2 and E6, 37.5 mm h^{-1}) and the third and fifth rainfall events (E3 and E5, 45 mm h^{-1}), the concentrations of the larger sediment sizes (C_5 - C_7) remain approximately equal for the same rainfall intensity on both the rising and falling limbs. In short, the measurements show that the larger particle size concentrations are determined primarily by the precipitation rate, not by the condition of the soil surface. This is likely due to the deposited layer already becoming dominated by the larger particles by the end of E1, thus the soil surface (the deposited layer) undergoes only relatively minor changes during subsequent events [Sander *et al.*, 2011; Kim and Ivanov, 2014].

2.3.1.3 Hysteresis loops

Figures 2.6 and 2.7 show plots of q versus sediment concentration for Collectors 1 and 2, respectively. Although minor differences can be seen, the results for the two collectors are similar. As described above, the maximum total sediment concentration occurs before the maximum discharge (Figures 2.3–2.5), thereby generating a clockwise hysteresis loop [Williams, 1989; Sander *et al.*, 2011]. As a result of compaction during the early rainfall events, as well as the initial removal of easily erodible fine sediment followed by the greater protection of the soil due to the growth of deposited layer with time, and the subsequent domination of this layer by the

larger size classes, less erosion takes place during the falling limb of discharge in comparison to the rising limb [Colby, 1963; Miller and Baharuddin, 1987; Sander *et al.*, 2011; Kim and Ivanov, 2014]. Clockwise hysteresis loops were measured for the three finest particle size classes (C_1 - C_3 , Figures 2.6 and 2.7). However, as the sediment particles become larger (C_4 - C_7), the hysteresis loops gradually become narrower and there is almost a linear relation between the discharge and sediment concentration, i.e., the hysteresis disappears. The concentration-discharge data show that, within a temporally varying rainfall event, different particle sizes not only have different hysteresis patterns, but also that their individual behaviors are strongly coupled. This is because the hysteresis patterns for the fine particles arise due to the presence of the larger particles within the original soil, which have a greater deposition rate compared to the finer particles. The larger particles with their nonhysteretic behavior limit the supply of the finer particles, resulting in hysteretic loops for these latter sediment sizes [Sander *et al.*, 2011].

The ploughed and smoothed flume surface initially provide a source of easily erodible sediment that has a greater availability of fine particles. During the first rainfall event, rapid increases in both C_1 and C_2 occur, with $C_2 > C_1$ as the mass proportion of sediment in class 2 is far greater than that in size class 1 ($p_2 \gg p_1$, Table 2.1). Since the deposition rate of suspended sediment is given by $v_i C_i$ in equation 1.3, the smaller classes contribute minimally to the deposited layer due to their low settling velocities. As the main source of the fine particles is the original soil, access to these particles becomes reduced by growth of the deposited layer (i.e., supply limited) [Parlange *et al.*, 1999; Bussi *et al.*, 2014]. Therefore, both C_1 and C_2 rapidly reduce from their initial peaks. Over successive events, the deposited layer becomes increasingly dominated by larger particles with any previously deposited fine material being slowly stripped out (Figure 2.8).

The availability of the finer particles reduces due to different factors. First, raindrops non preferentially eject sediment from the bed [Legout *et al.*, 2005]. Suspended transport of finer sediment sizes, however, occurs preferentially, as does deposition of larger sediment sizes. The combination of these two effects leads to a surface soil layer that is progressively denuded of finer sediments as they are transported from the flume. Other things being equal, this means that the sediment concentrations for finer size classes leaving the flume will be reduced. Second, soil compaction decreases the depth of the soil into which the raindrops penetrate, i.e., less soil is able to be detached by the raindrop impact. Consequently, during the falling limb of the

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

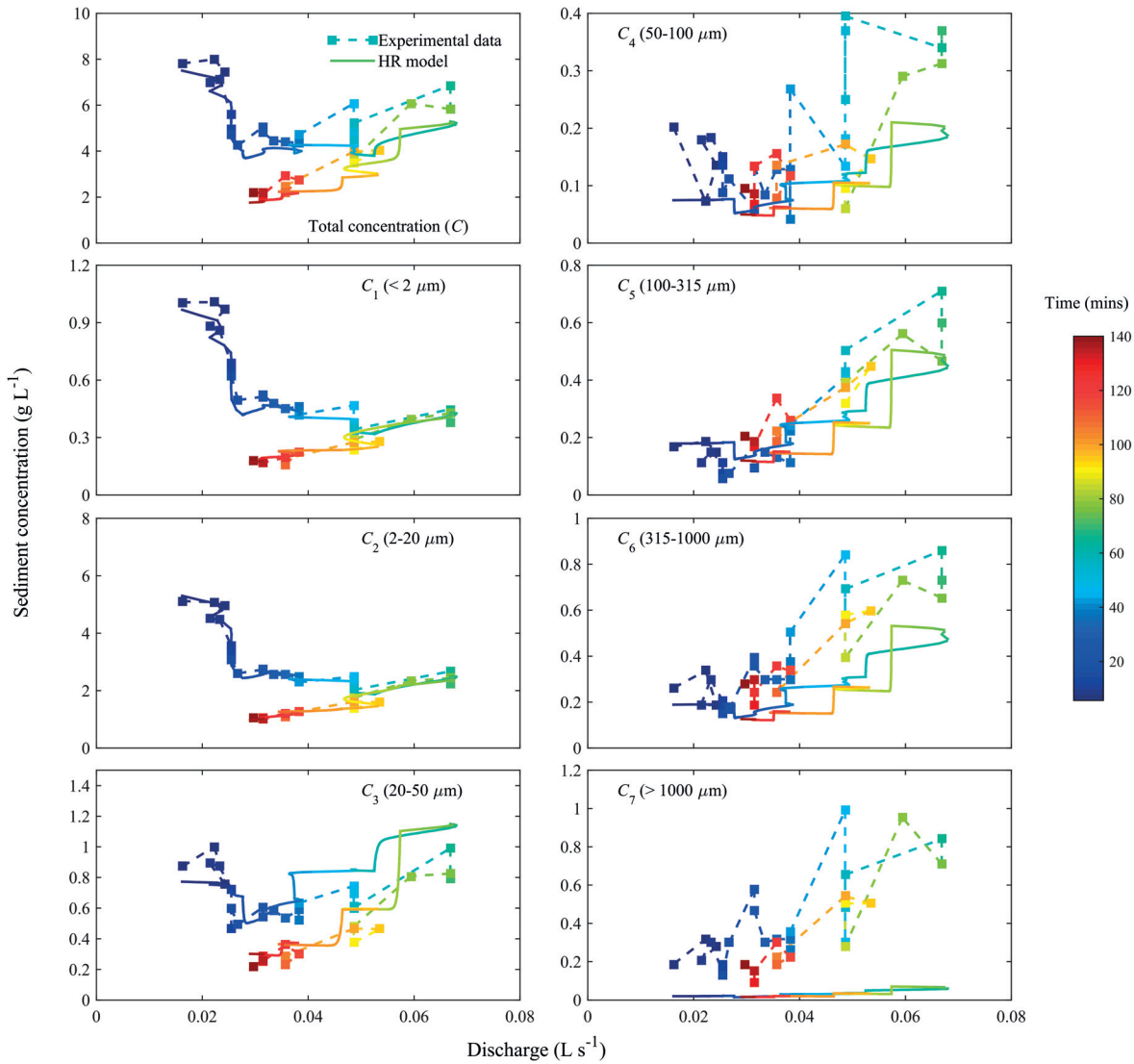


Figure 2.6 – Hysteresis loops (measured data and HR model results) associated with the total sediment concentration (C) and individual particle size classes ($C_1 - C_7$) (Collector 1). The elapsed time (0-140 min) is shown by the color bar.

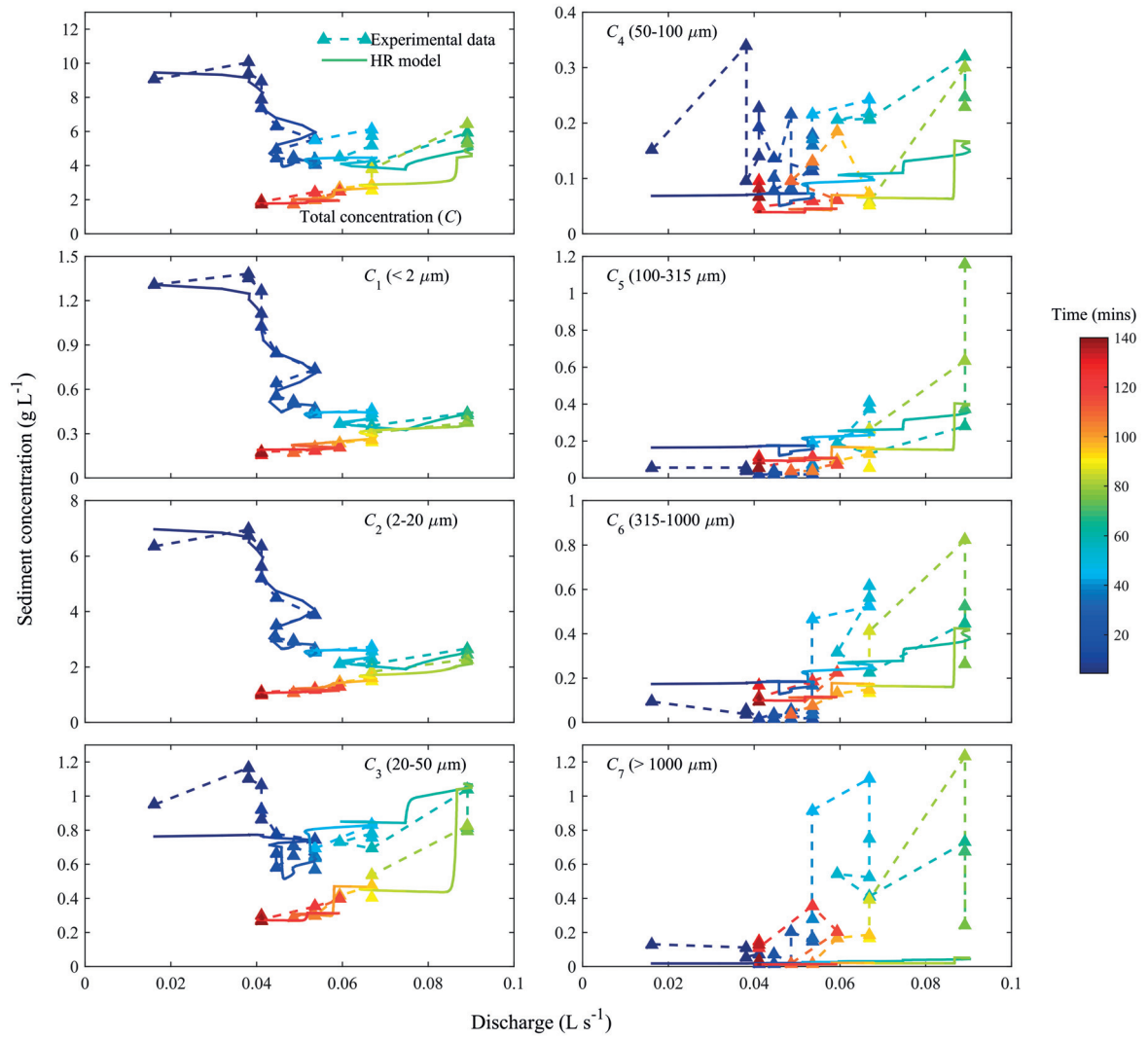


Figure 2.7 – Hysteresis loops (measured data and HR model results) associated with the total sediment concentration (C) and individual particle size classes ($C_1 - C_7$) (Collector 2). The elapsed time (0-140 min) is shown by the color bar.

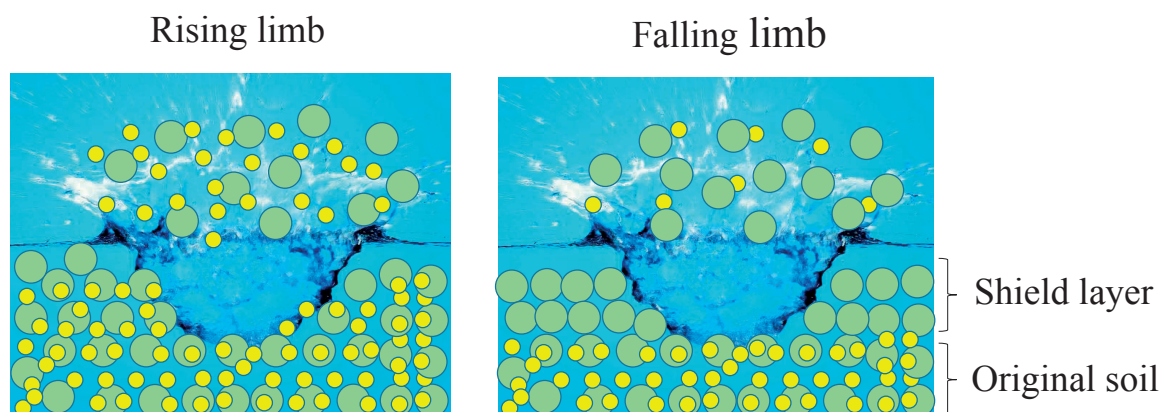


Figure 2.8 – Schematic depiction of raindrop impact on the soil surface. Finer particles are available for suspension due to raindrop impact on the rising limb. Over time, the finer particles are preferentially removed, leaving a shield layer composed mainly of larger particle sizes [Colby, 1963].

hydrograph for the same discharge, lower sediment concentrations are observed since it is now the larger, deposited particles that are being eroded. That the larger particles are seen to have similar concentrations on both the rising and falling limbs are due to their consistent availability (Figure 2.9) in the deposited layer. For this situation, changes in concentration between events are related to the rainfall intensity, i.e., detachment limited. Thus, the different behavior across the size classes highlights the importance of the particle size distribution in the development of hysteresis loops. These results reinforce the findings of *Bussi et al.* [2014] for the Goodwin Creek catchment and the numerical study of *Kim and Ivanov* [2014] on the role of initial loose sediment (equivalent to the deposited layer in the HR model) in determining sediment transport patterns. That is, the initial spatial distribution and sediment size class composition of the deposited layer play an important role in determining the different types and orientations of sediment hysteresis loops [Sander et al., 2011; Zhong, 2013; Bussi et al., 2014].

Hysteresis in total sediment concentration-discharge plots for river flow was analyzed by *Williams* [1989] who explained clockwise loops as being due to either “a depletion of available sediment before water discharge has peaked” or to “the formation of an armored layer prior to the occurrence of the discharge peak.” The results presented

here on the sediment size classes composing the soil highlight the important role of the reduction of fines within the surface soil layer—this reduction over the course of the experiment underpins the observed hysteresis. At the same time, the surface layer protects the underlying soil, so in that sense it is an “armored layer,” i.e., both reasons given by *Williams* [1989] apply to our experiments.

2.3.2 Model

2.3.2.1 Calibrated parameters

We return to the optimized parameters for the HR model given in Table 2.2. The results show that the maximum detachability (a) occurs in the first rainfall event (E1). This value reduces during E2, and then varies little. Likewise, the value of critical mass (m^*) is a minimum for the first rainfall event, varying little thereafter. These results reflect the differences in the soil structure (i.e., soil compaction) that had a significant effect during the first rainfall event, but was negligible for later events.

Another factor is the water layer depth (D) that protects the soil from raindrop splash erosion (equation 1.3). The modeled water depths reported in Table 2.2 increase as expected with rainfall intensity, but also because soil compaction during the falling limb decreases the infiltration rate and smooth the soil surface [*Jomaa et al.*, 2013], D tends to be lower on the falling limb of the hydrograph for the same rainfall intensity event.

The mass of deposited sediment required for complete shielding (m^*) depends on both the rainfall rate and flow depth. For the same rainfall rate and soil condition but for a lower water depth, raindrops are able to penetrate a greater distance into the soil surface [*Hairsine et al.*, 1999; *Jomaa et al.*, 2012b]. Hence, a greater thickness, or increased mass m^* , is required for the deposited layer to absorb the raindrop energy and fully protect the underlying soil. Consistent with this observation, the results of Table 2.2 also show that, at different rainfall rates, higher values of m^* are predicted for Collector 2 on which the water layer (D) is less than for Collector 1.

The match of the model with the experimental results is good for the total sediment concentration and for the finer particle size classes, with poorer matches for the larger particles (Figures 2.4–2.7). For the largest particle size class (C_7), although the trend of the model is correct (i.e., rising and falling sediment concentrations depending

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

on the precipitation rate), the modeled concentrations of the large particles are less than measured in the experiment. As noted previously [Jomaa *et al.*, 2012b, 2013], in the HR model, the larger particles are assumed to be transported as suspended load that is rapidly deposited due to their high settling velocities. However, large particles are more likely transported due to rolling, saltation and ejection during the experiment, which would account for the higher measured values compared to the model predictions. A caveat on the HR model application is that, although the results are consistent with the experimental data, the model was not used to predict the experimental measurements, rather the model was calibrated to them. Our purpose here was not prediction, but to test whether the HR model could reproduce the time-dependent size class hysteretic behavior observed in the experimental results in a physically consistent manner.

2.3.2.2 Deposited layer mass

The total mass of the shield layer and its sediment-size composition is given by the calibrated HR model. In Figure 2.9, model results for the total mass of the deposited layer (m) and the proportion of seven particle size classes (m_1 - m_7) are presented for Collector 1 only as results for Collector 2 are similar. At the start of the first rainfall event, the initially ploughed and smoothed surface easily erodes, resulting in a rapid increase in the suspended concentration (Figure 2.4), which is immediately followed by deposition and a rapid increase of the deposited layer (Figure 2.9). Initially, the deposited layer contains the smaller sediment size classes (C_1 - C_3) even though they have relatively low settling velocities. This is because the deposition rate (as given by $d_i = v_i C_i$ in equations 1.2 and 1.3) shows that a high suspended sediment concentration can compensate for a low settling velocity to still give significant deposition of small particles. However, as time increases, the supply of the smaller particles (C_1 - C_3) reduces as the original soil becomes protected, and the smaller particles within the deposited layer are gradually redetached and advected downstream to the flume exit, resulting in the increased contribution of all larger sized particles, i.e., sizes C_4 through C_7 (Figure 2.9) [Heilig *et al.*, 2001; Salant *et al.*, 2008; Tromp-van Meerveld *et al.*, 2008; Sander *et al.*, 2011; Kim and Ivanov, 2014]. The increased rainfall at the start of the second event (E2) resulted in little change in the flow depth (Table 2.2). At the same time, the sediment detachment rate increased,

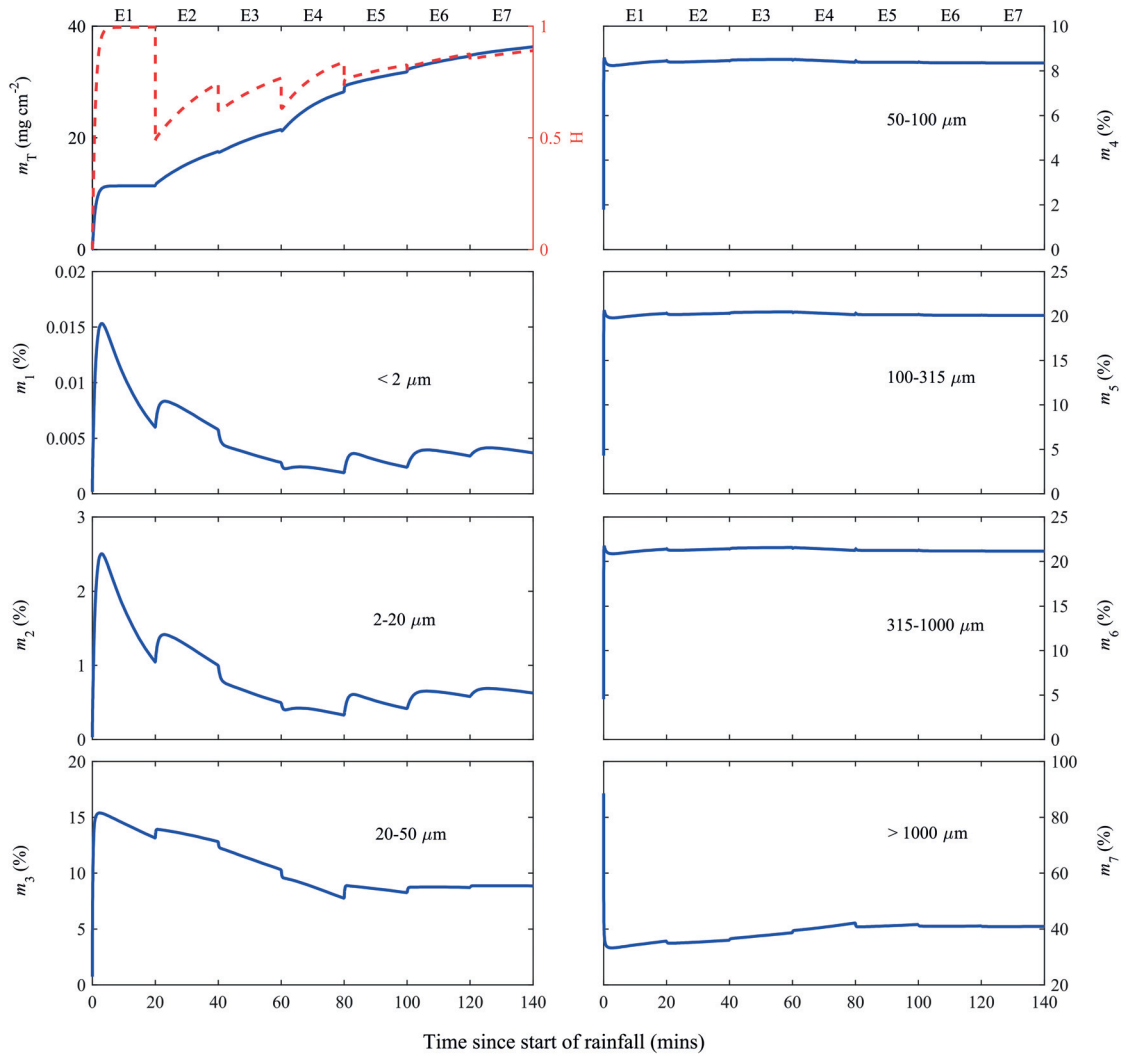


Figure 2.9 – HR model predictions of the total mass of the deposited layer (m) and the contribution of seven different particle size classes ($m_1 - m_7$) within the shield layer (Collector 1). The variation of degree of shielding (H) is shown along with the total mass.

Chapter 2. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects

resulting in an increased value of m^* to protect the original soil and a corresponding sudden decrease in H (equation 1.3). Thus, the increased rainfall rate results in increased penetration of raindrops through the depositing layer to access a greater amount of fine particles. For m_1 - m_3 (Figures 2.8 and 2.9), this increase reflects the behavior seen at the beginning of the first event, but at a lower magnitude due to raindrop compaction of the soil. Simultaneously, there are small increases in the measured sediment concentrations in the effluent for size classes C_1 - C_3 (Figures 2.4 and 2.5). Again, there is a rise in the contributions of the larger particles (m_4 - m_7) to the deposited layer (Figure 2.9, E2). From event E3 onward, there is a continual removal of the small size classes (m_1 - m_3) that is accompanied by growth in the contribution of the largest size class, while that of classes m_4 , m_5 and m_6 remain relatively static. By this time, the deposited layer has become so dominated by the largest size classes that changes in rainfall rate (Figure 2.2) and flow rate (Figure 2.3) have only a minor impact on it.

2.4 Concluding remarks

Sediment transport as a result of multiple continual rainfall events was studied via experiments and modeling. Specifically, we investigated the hysteresis loop patterns of different sediment size classes versus flume discharge for time-varying precipitation rates. To this end, seven consecutive rainfall events were applied to an initially ploughed and smoothed soil. Sediment concentrations at the flume exit were taken from samples from two collectors. The results were further analyzed by calibrating the HR model to the measurements. We examined the behavior of the different particle size classes during the multiple rainfall events in the absence of rills.

For an initially dry and ploughed soil, clockwise hysteresis loops in sediment concentration versus discharge rate were generated for the total sediment concentration and the concentrations of the finest particle size classes. In contrast, for the larger particle sizes, the hysteresis loops are narrower and have a more irregular shape. Indeed, it is not clear whether they exhibit hysteresis. The results suggest that the contribution of the finer particles to the total eroded mass reduces over the course of the experiment, independent of the precipitation rate (source-limited delivery). However, the contribution of larger particles to the total eroded mass increased during

the rising limb and decreased during the falling limb, reflecting transport-limited behavior of the larger size classes. We remark that soil compaction is not necessary for the appearance of hysteresis. Instead, at least for our experiments, the key factor is the decreasing availability of finer sediments. The combination of deposited material and compaction also protects the original soil from raindrop erosion. Another factor affecting raindrop erosion is surface water, since increasing water depths protect the soil. The average surface water depth changes due to compaction and the variable precipitation rate. Consistent with the low precipitation rates applied during most of the experiment, modeled water depths were small according to the calibrated HR model (Table 2.2), and which we noted visually. Consequently, it is unlikely that water depth played a significant role protecting the soil in the results reported here.

A crucial feature of the HR model is its ability to simulate the sediment size distribution in the deposited layer (i.e., deposition of previously eroded sediment), which provides a quantitative basis for simulating hysteresis in the discharge-sediment concentration plots. The model results show that the reduction in the availability of the finer sediment sizes in the deposited layer results in hysteresis in the total sediment concentration plots, as well as in the finer size classes. Additionally, the model results are consistent with the reduction (even absence) of hysteresis evident in the experimental data for the larger sediment size classes.

However, the just-mentioned reduction of the proportion of fine sediment sizes is not essential for hysteresis to occur. To be clear, the distribution of size classes plays an important role in determining the shape and magnitude of the hysteresis loop, but it is not the controlling factor. In the extreme case of a soil that is composed of only a single size class, simulations with the HR model show that clockwise, anticlockwise, and figure eight loops can still be obtained. As found by *Sander et al.* [2011], the shapes of the hysteresis loops produced are still dependent on the initial condition of the deposited layer, i.e., on the initial availability of easily erodible sediment. These shapes are dependent on particle size. That is, as the (single) particle size increases, the size of the loop diminishes, and hysteresis effectively ceases, in agreement with the behavior of the largest size classes shown in Figures 2.6 and 2.7.

3 Statistical characteristics of overland flow morphologies

Mohsen Cheraghi¹, Andrea Rinaldo^{2,3}, Graham C. Sander⁴, Paolo Perona⁵ and D. A. Barry¹

¹Ecological Engineering Laboratory (ECOL), Institute of Environmental Engineering (IIE), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

²Ecohydrology Laboratory (ECHO), Institute of Environmental Engineering (IIE), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

³Dipartimento di Ingegneria Civile Edile e Ambientale, Università di Padova, Padua, Italy

⁴Department of Civil and Building Engineering, Loughborough University, Loughborough, UK

⁵Institute for Infrastructure and Environment, School of Engineering, The University of Edinburgh, Edinburgh, UK

Published in Geophysical Research Letters:

Cheraghi, M., Rinaldo, A., Sander, G. C., Perona, P. and Barry, D. A. (2018). Catchment Drainage Network Scaling Laws Found Experimentally in Overland Flow Morphologies. *Geophysical Research Letters*, 45, 9614-9622. <https://doi.org/10.1029/2018GL078351>.

3.1 Introduction

Even with markedly different environmental and geological conditions, catchment drainage networks have similar geometrical characteristics that take the form of power laws [Rodríguez-Iturbe and Rinaldo, 1997; Rinaldo et al., 2014], as measured for different areas [Hack, 1957; Mandelbrot, 1982; Tarboton et al., 1989; Rigon et al., 1996]. Hack's law [Hack, 1957] states that the upstream length (l , the longest flow path into each point) and drainage area (A) are related via a scaling relation ($l = A^h$) where the exponent h was measured in the range of [0.5-0.7] for different river networks [Hack, 1957; Gray, 1961; Mueller, 1972; Mosley and Parker, 1973; Montgomery and Dietrich, 1992; Maritan et al., 1996; Rigon et al., 1996, 1998], with an average value of about 0.58 [Willemín, 2000]. Also, for the fluvial parts of landscapes, power-law relations with exponent ranges of [0.42-0.45] and [0.5-0.9] were observed for the exceedance probabilities of drainage area and length, respectively [Rodríguez-Iturbe and Rinaldo, 1997; Rigon et al., 1996; Crave and Davy, 1997; Paik and Kumar, 2011]. Different explanations of these power laws are available [Banavar et al., 1999; Dodds and Rothman, 2000; Birnir et al., 2001; Banavar et al., 2001; Birnir et al., 2007; Birnir, 2008; Rinaldo et al., 2014], including self-organized dynamic systems [Bak et al., 1988; Rinaldo et al., 1993; Frigg, 2003; Marković and Gros, 2014; Watkins et al., 2016], invasion percolation [Stark, 1991; Hunt, 2016] and minimum energy dissipation [Rodríguez-Iturbe et al., 1992].

Catchment drainage networks are essentially static structures in the landscape, i.e., their temporal evolution cannot be readily measured. On the other hand, experimental geomorphology has a longstanding tradition [e.g., Schumm and Khan, 1971; Flint, 1973; Mosley and Parker, 1973; Parker, 1977] and permits detailed and rapid investigations of changes in surface morphology due to rainfall or overland flow aggregation [e.g., Crave et al., 2000; Brunton and Bryan, 2000; Römkens et al., 2002; Hasbargen and Paola, 2003; Gómez et al., 2003; Pelletier, 2003; Turowski et al., 2006; Babault et al., 2007; Yao et al., 2008; Tatard et al., 2008; Paola et al., 2009; Bonnet, 2009; Berger et al., 2010; Graveleau et al., 2012; Rohais et al., 2012; McGuire et al., 2013; Reinhardt and Ellis, 2015; Sweeney et al., 2015]. For instance, dynamic changes of a rill network in uncohesive sediment under a constant uplift rate were observed by Hasbargen and Paola [2000]. In contrast, rill networks in a cohesive sediment evolved along the previously generated rills [Bennett and Liu, 2015] due to surface resistance. Singh et al. [2015] generated rill networks in a 0.5-m×0.5-m experiment under spatially

uniform but temporally variable rainfall and constant uplift rate. They found that the drainage area distribution was described by a power law with an exponent of 0.5. Similarly, *Bennett and Liu* [2015] examined rill formation at the flume scale (7 m×2.4 m) and found an exponent of about 0.5 for Hack's law.

In summary, geometrical characteristics of catchment drainage networks have a high degree of similarity. These same characteristics are evident in channeled surfaces in laboratory studies. Here we extend these studies by considering the flow network on an unchanneled sediment. Specifically, we measured the surface evolution of an unconsolidated sediment under non-uniform rainfall and overland flow such that no (observable) rills were formed. However, the surface roughness produces a drainage network representation of the overland flow, which is then subjected to geometrical analysis.

3.2 Experiment

We used a 2-m × 1-m erosion flume as shown in Figure 3.1. The total sand depth was 15 cm. Uncohesive sediments with a mean diameter of 0.53 mm (Table 3.1 and Figure 3.2) were placed in three successive 5-cm layers within the flume. Each layer was compacted by 30-min rainfall (droplet size of 3-7 mm) with an intensity of 10 mm h⁻¹ followed by consolidation via a 600 kg m⁻² weight. Then, the sediment was air dried for 48 h. Heterogeneous rainfall was generated by two sprinklers located 3 m above the sediment surface. The distribution of rainfall intensity is shown in Figure 3.4h. The average rainfall was 85 mm h⁻¹ with a Christiansen uniformity coefficient [*Christiansen*, 1942] of 26%. The rainfall was applied continuously except a 30-min break for each laser scan. There was no drainage from the flume bottom and all surface flow was collected at a single, 4-cm wide outlet, located at 6 cm above the base of the flume. The experiment started with a smooth surface. At the outlet, the initial elevation difference between the sediment surface and the base level was 9 cm. The flow rate at the flume outlet is plotted in Figure 3.3. During the first 5 min, the sediments became saturated and the outlet flow rate was low. Afterwards, there was a rapid elevation drop near the outlet (blue areas in Figure 3.4a) until about $t = 10$ min. Until this time, it was not possible to capture the flow rate due to high sediment concentrations at the outlet. Scanning started after 15 min when the flow rate became steady. At this stage, the sediments were saturated and the precipitation

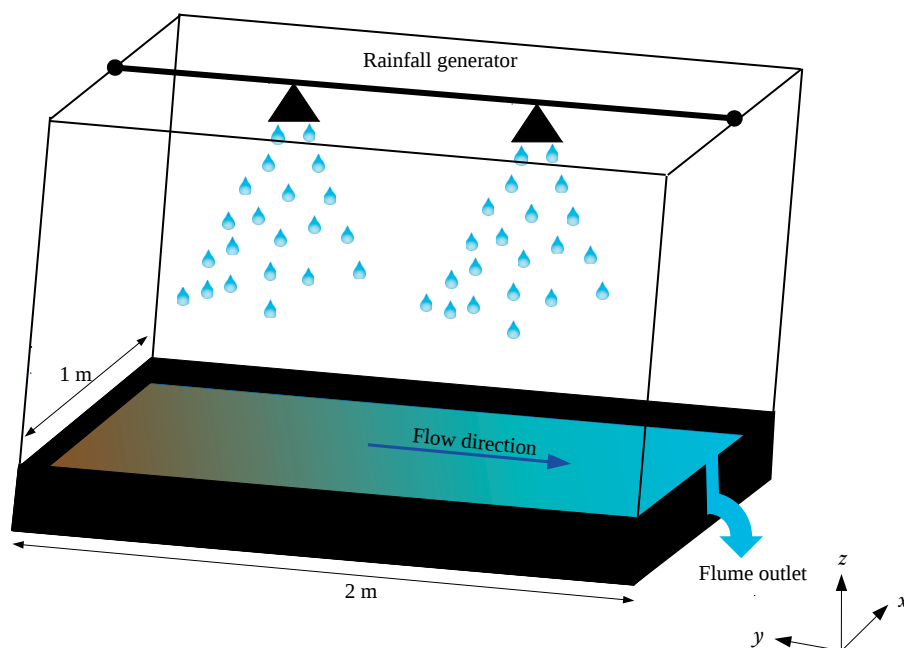


Figure 3.1 – Schematic of the flume experiment. The flume width, W , was 1 m and its length, L , was 2 m. The sediment vertical elevation (z) was measured from the outlet ($z = 0$). The relief height (the maximum elevation) was 0.19 m. Each layer was compacted by 30-min rainfall (droplet size of 3-7 mm) with an intensity of 10 mm h^{-1} followed by consolidation via a 600 kg m^{-2} weight. Then, the sediment was air dried for 48 h.

and discharge rates were equal. A 3D laser scanner (Konica Minolta Vivid 910), with about 4-mm horizontal resolution and accuracy of 0.1 mm in the vertical direction, was used to extract Digital Elevation Models (DEMs) at 0.25, 0.5, 1, 2, 4, 8 and 16 h. The scanner was calibrated using 20 fixed points on two bars along the flume's lateral walls. Eight individual scans were taken to cover the entire flume. Following registration and post-processing, each DEM produced was trimmed 2 cm from the side walls and 10 cm from the upstream wall. The surface morphology data are available at: <http://doi.org/10.5281/zenodo.1292113>.

Table 3.1 – Characteristics of the sand used in the experiment.

Bulk density	Particle size (d) range	d_{50}	$d < 0.6 \text{ mm}$	$d > 2 \text{ mm}$
$1584 \text{ (kg m}^{-3}\text{)}$	0-6 mm	0.53 mm	70 (% mass)	12 (% mass)

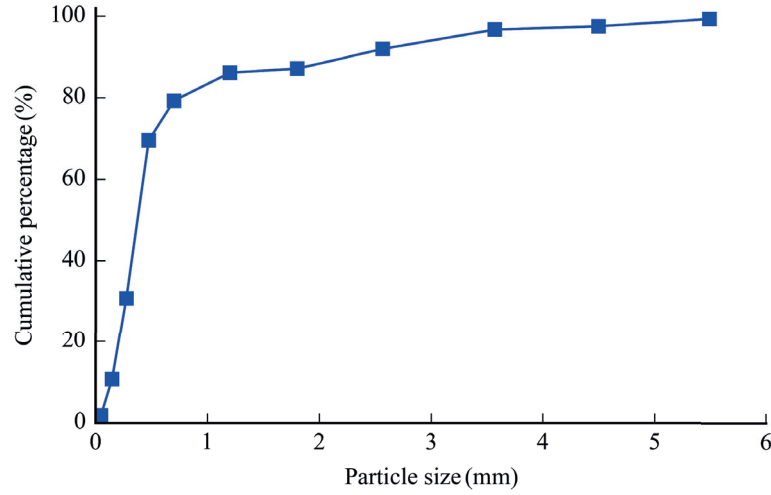


Figure 3.2 – Particle size distribution of the sediment.

3.3 Results and discussion

3.3.1 Morphology and drainage network evolution

The elevation change during the experiment is shown Figure 3.4. The sediment elevation was measured from the outlet ($z = 0$). For convenience, we refer to the ranges $z \leq 60$ mm and $z \geq 60$ mm as the downstream and upstream, respectively. Overall, the morphology evolution can be divided into two steps: (i) until $t = 4$ h, most of the variation occurred at the upstream end while the downstream end did not show any considerable evolution, and (ii) after $t = 4$ h, the downstream morphology propagates into the upstream.

To characterize the morphology, a network was generated based on the measured surface scans (Figure 3.4a-g) and precipitation (Figure 3.4h). Pit points were removed following *Planchon and Darboux* [2002]. Similarly to large scale river networks, the discharge distributions (Q) and drainage area (A) are computed via the D8 algorithm [O'Callaghan and Mark, 1984]:

$$Q_i = \sum_{j=1}^8 w_{ji} Q_j + R_i \Delta x \Delta y \quad (3.1)$$

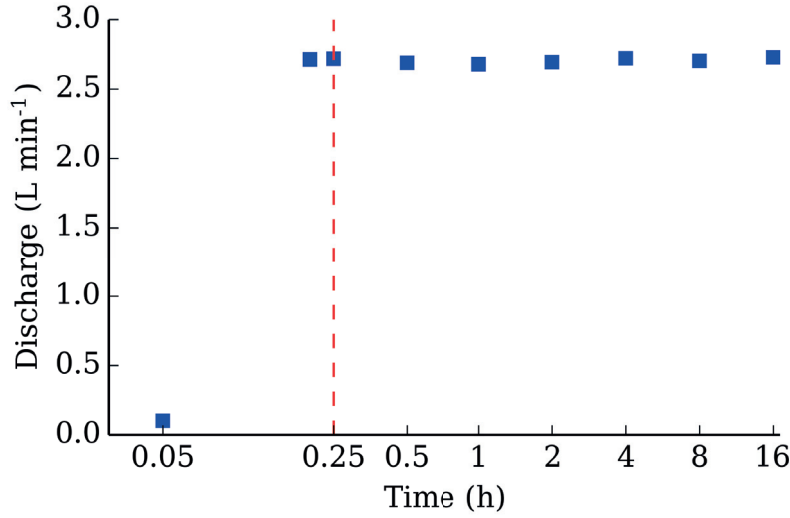


Figure 3.3 – Measured water flow rate at the flume outlet.

$$A_i = \sum_{j=1}^8 w_{ji} A_j + \Delta x \Delta y \quad (3.2)$$

where the summation over j refers to the eight cells surrounding the i th cell. The slopes from each cell (i) into each of the eight neighbor cells (j) were calculated, with flow directed along the steepest descent. The value of w_{ji} is unity if the cell j flows into cell i , otherwise it is zero. R_i (mm h⁻¹) is rainfall intensity at cell i (Figure 3.4 h) and Δx (mm) and Δy (mm) are the grid sizes in x and y directions, respectively.

The distribution of drainage area and discharge at different times are plotted in Figures 3.5 and 3.6, respectively. At $t = 0.25$ h (Figure 3.5a), four separate branches depicted by A–D drained into the flume's outlet ($x = 0$, $y = 0$). Then, at $t = 0.5$ h (Figure 3.5b), branch C joined B and branch BC was generated while a minor change in the network was evident in the upper part of the network. After 1 h (Figure 3.5c), junction A became attached to BC and the pathway denoted ABC was formed. At $t = 2$ h (Figure 3.5d), the area drained by ABC inclined to the right side. Furthermore, branch D drained a greater proportion of the precipitation as it assumed part of the upstream area previously drained by ABC. Finally at $t = 4$ h, the network ABCD was generated (Figure 3.5e). At later times ($t = 8$ h & 16 h), the high flow part of ABCD became more dominant and moved to the right (Figure 3.5f & g). Variations in the drainage area and discharge network mostly occurred in the first 8 h of the

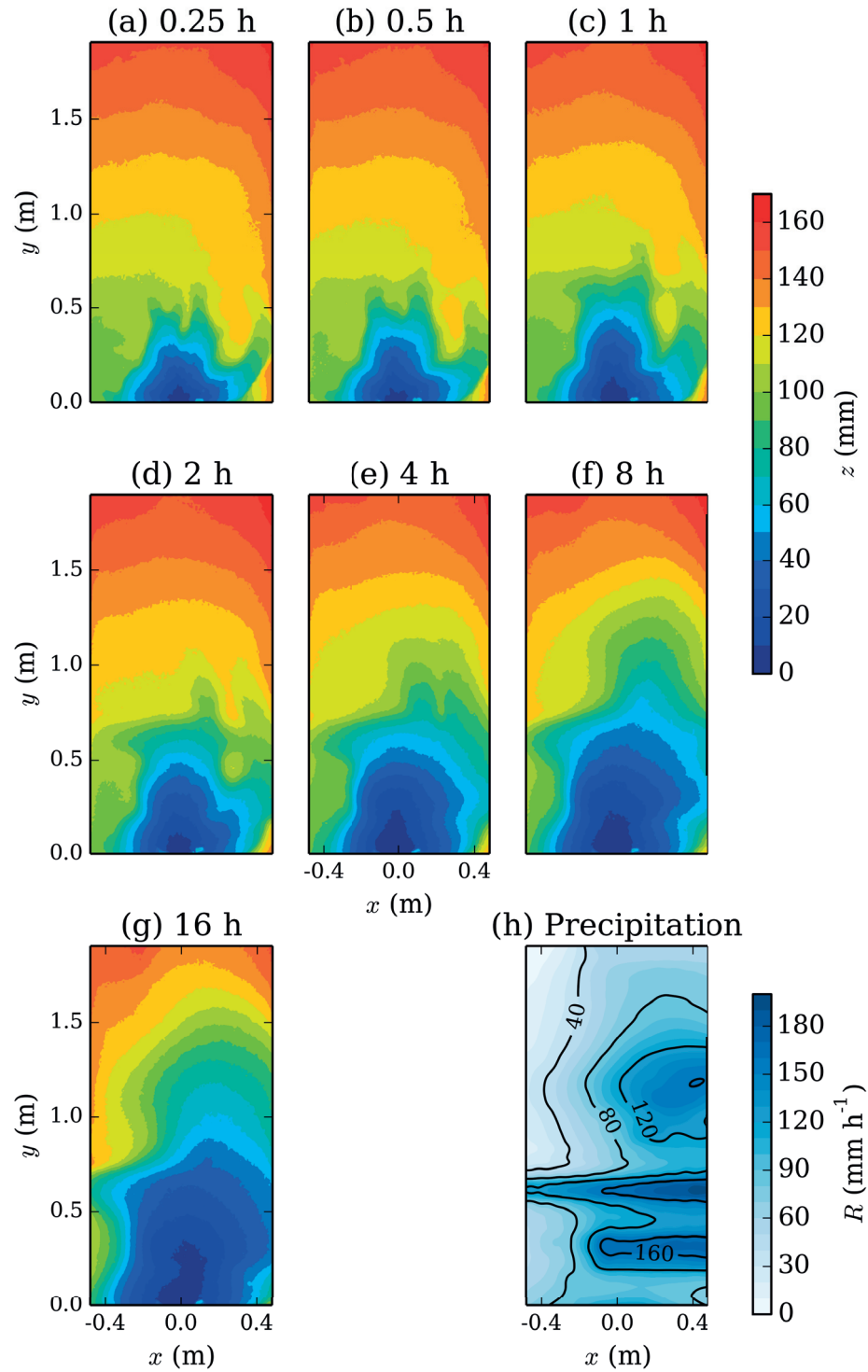


Figure 3.4 – Measured morphology evolution during the 16-h experiment. Due to the non-uniform rainfall, the morphology changes increase from the left side (low precipitation rate area) towards the right (high precipitation rate area).

experiment, similarly to the surface morphology. Changes were less rapid in the second 8 h, although the main structure of the network was reinforced and some local changes to the low-order pathways took place. The evolution of the downstream (Figure 3.4e) started at the same time as the network (ABCD) was generated at $t = 4$ h (Figure 3.5e). The network's width function was computed for each scan to quantify its temporal evolution (Figure 3.7).

Even though the flow covers the entire surface and is continuous (except perhaps for raindrop impacts), the D8 algorithm leads to its description as a network, which was considerably reorganized during the 16-h rainfall duration (Figure 3.5). We recall that these networks do not represent observable surface rills, but rather the drainage network derived from the surface morphology as captured by the surface scans. As shown in Figure 3.8, due to shorter erosion time scales, the fine sediment particles are rapidly removed while the larger particles move slowly down the surface [Polyakov and Nearing, 2003; Sander *et al.*, 2011; Wang *et al.*, 2014; Kim and Ivanov, 2014; Lisle *et al.*, 2017] or are not moved at all, resulting in a surface partially covered by motionless pebbles. Therefore, the network evolution is a result of size-dependent sediment particle transport and raindrop-driven rearrangement on the surface.

3.3.2 Catchment scaling laws in overland flow morphologies

We next examine the statistical characteristics of the network. We first consider Hack's law [Hack, 1957], which is a well-known metric used in analyses of large scale river networks [Maritan *et al.*, 1996; Rigon *et al.*, 1996; Dodds and Rothman, 2001b]. For our case, the $A-l$ distribution is divided into 20 bins on a logarithmic scale. For each bin, the ratio between consecutive average moments of length were calculated. The results are plotted in Figure 3.9 for the first to fourth moments of l ($n = 1, 2, 3, 4$). They show a validation of a finite-size scaling framework for the distributions of l , in the form of $p(l) = l^{-\xi} F(l/A^h)$ where $F(x) \rightarrow 0$ for $x \rightarrow \infty$ and $F(x) \rightarrow 0$ for $x \rightarrow 0$, analogous to large scale river networks [Rigon *et al.*, 1996]. The power law relation is maintained for at least two orders of magnitude, with the scaling exponent h in the range of [0.54-0.6]. Upper and lower cutoffs affecting the scaling range were expected. Lower cutoffs are basically the limits of detectability. Upper cutoffs are associated with the maximum cumulative area or flow rate [Rigon *et al.*, 1996]. Another experiment at 10% slope with an average rainfall of 60 mm h^{-1} (Figure 3.16) showed a range of [0.51-0.55] for the

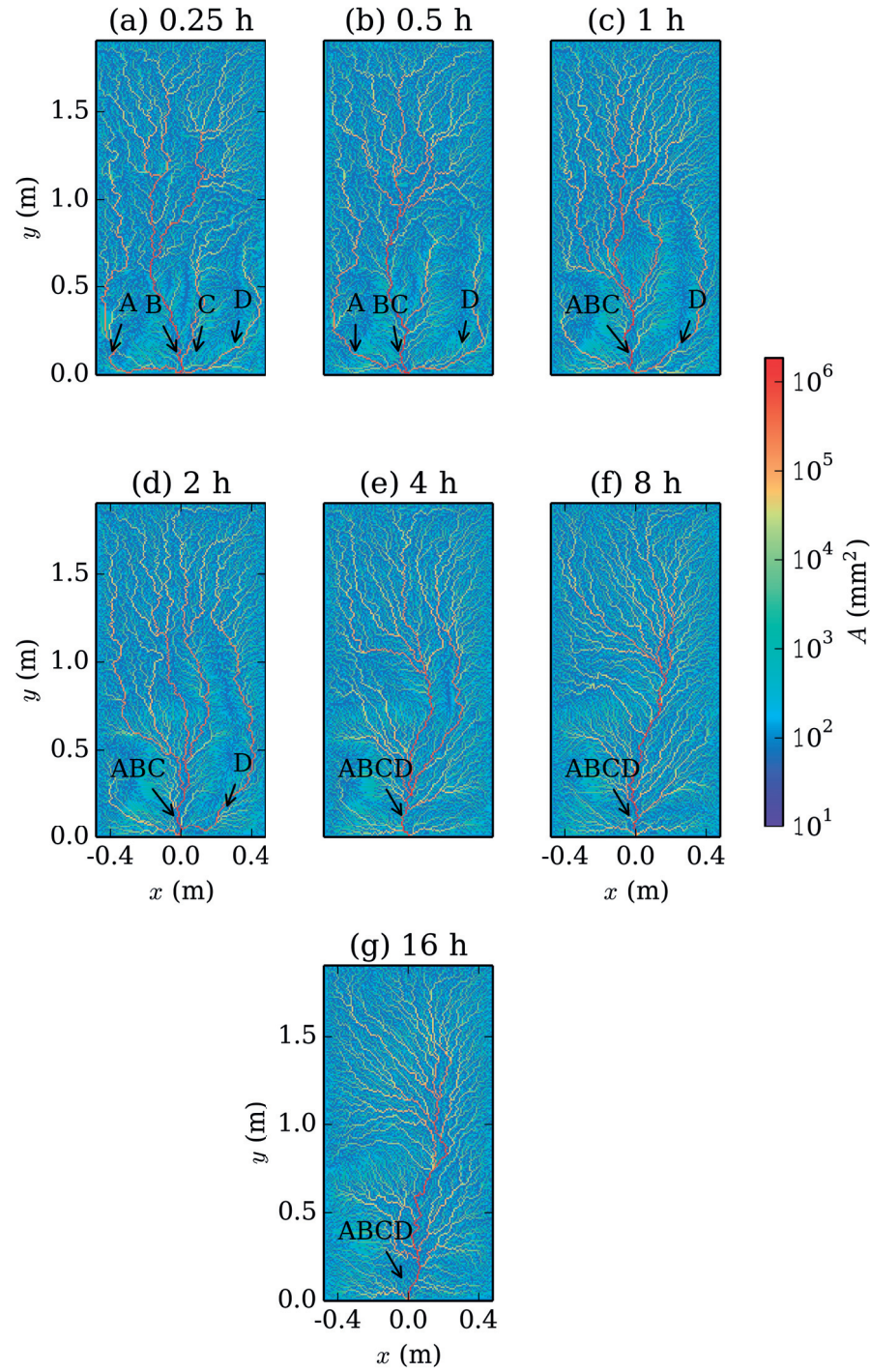


Figure 3.5 – Drainage area (A) distribution determined using the D8 algorithm and the measured morphologies shown in Figure 3.4a-g. Initially, the flow paths, e.g., at $t = 0.25$ and 0.5 h, reflect the initial surface condition and central drainage point at the flume exit. The impact of the higher-intensity rainfall on the right side of the flume is manifested in the main flow path, which increasingly moves to the right side over time (more details given in the text).

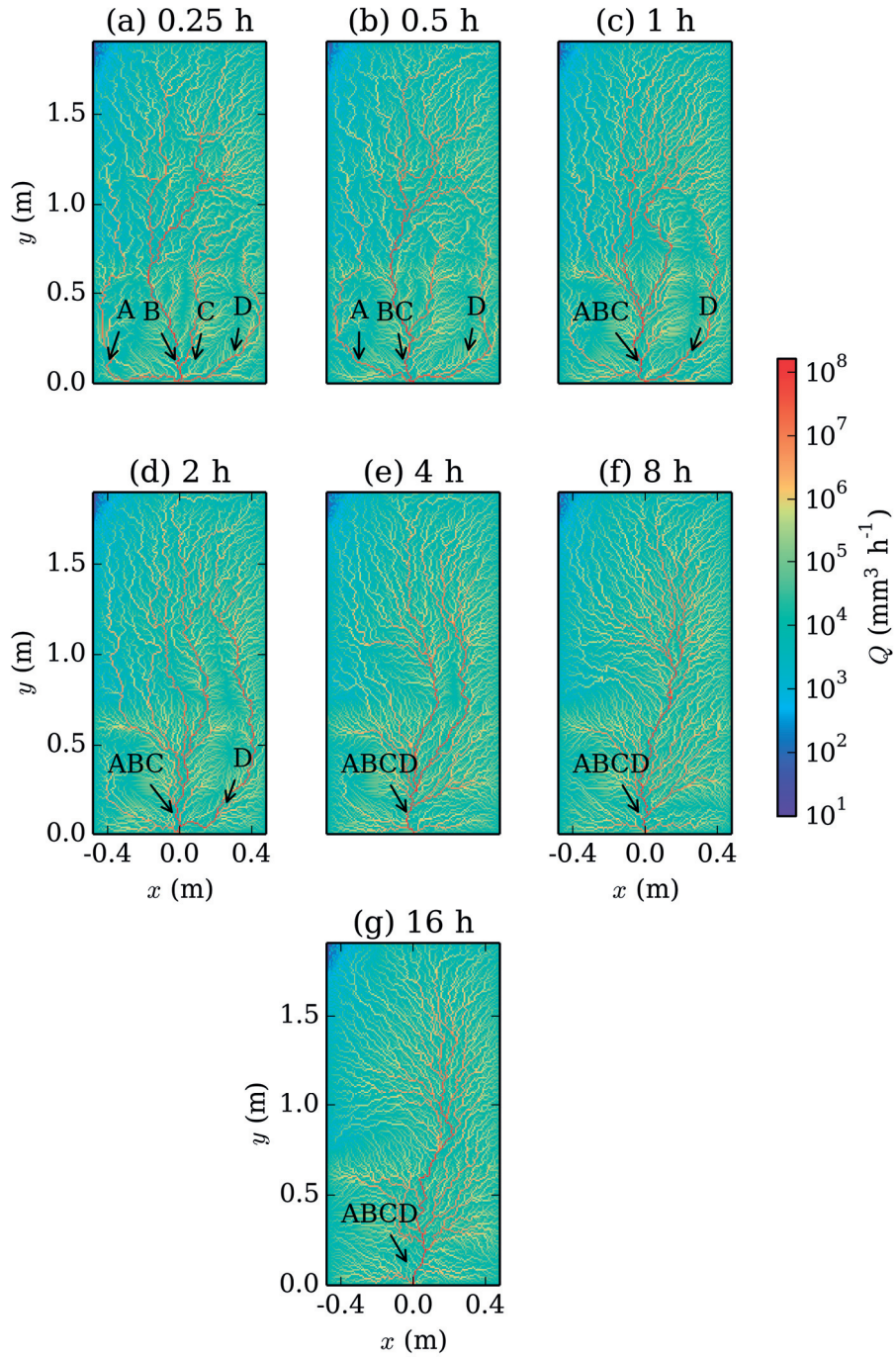


Figure 3.6 – Drainage discharge distribution determined using the D8 algorithm and the measured morphologies and the rainfall intensity (Figure 3.4). In spite of the heterogeneous rainfall intensity, the generated network and its evolution are very similar to the drainage area network (Figure 3.5). As expected, the maximum calculated discharge at the outlet (2.709 L min^{-1}) is consistent with the measured value at the steady state (Figure 3.3).

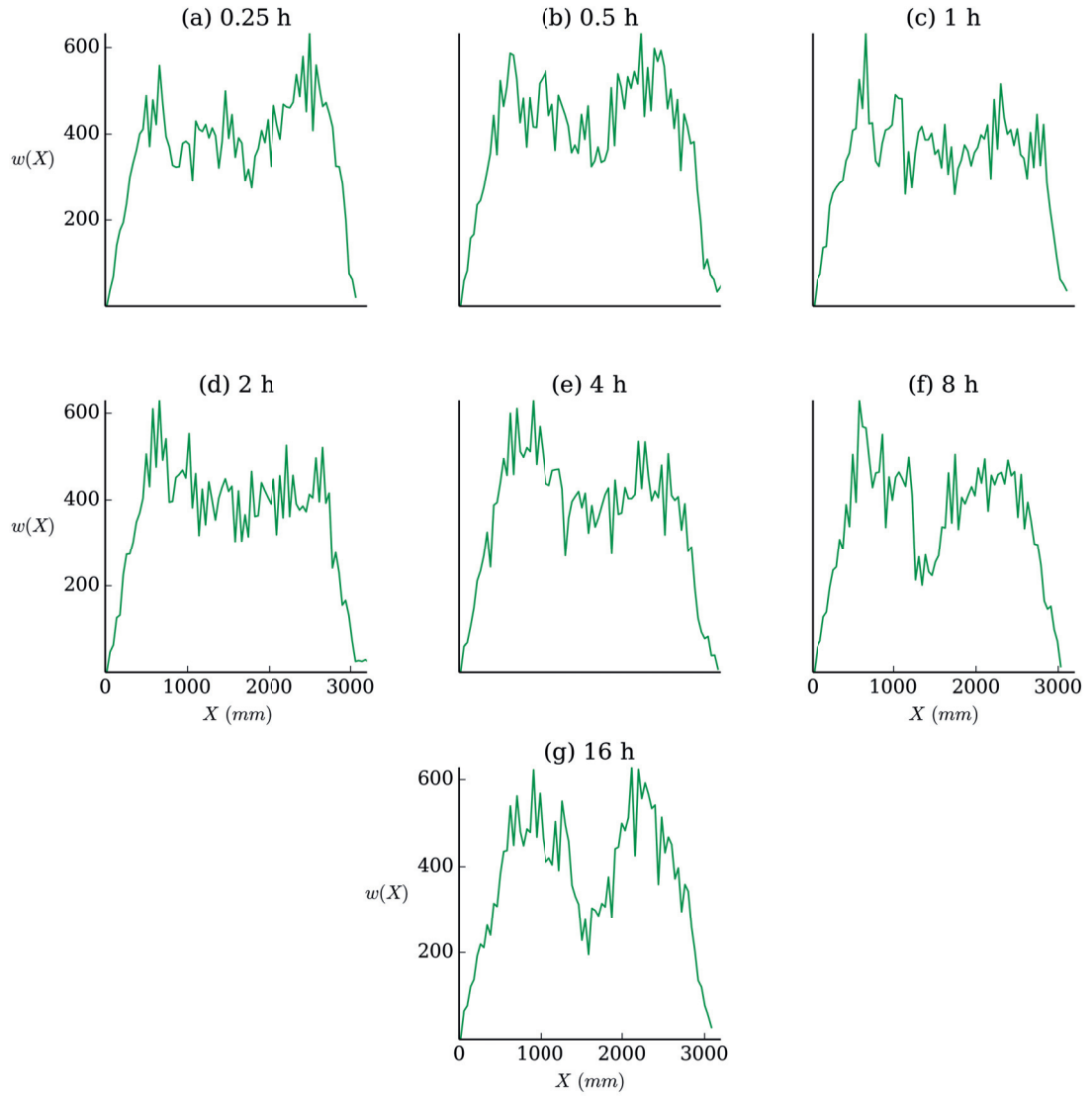


Figure 3.7 – Width function (w) of the drainage area (and discharge) network (Figures 3.6 and 3.5). The value X is the distance of each point from the flume outlet (along the flow paths of the network) and $w(X)$ is the number of points with the same distance (X) from the network outlet. The width function of the network dynamically changed during the morphology evolution.

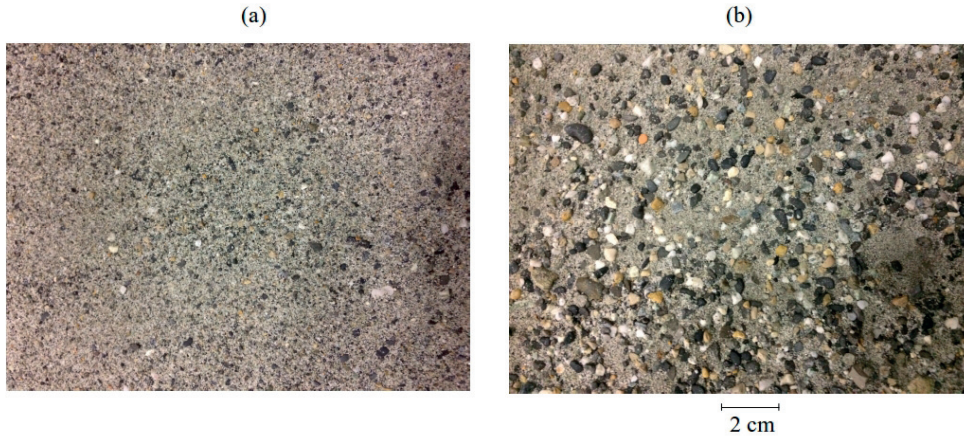


Figure 3.8 – Sediment surface at $t = 0$ (a) and $t = 16$ h (b).

Hack exponent. For both experiments, the Hack exponents agree with those found for large scale river networks [Hack, 1957; Gray, 1961; Mueller, 1972; Mosley and Parker, 1973; Mueller, 1973; Montgomery and Dietrich, 1992; Maritan *et al.*, 1996; Rigon *et al.*, 1996, 1998], which are in the range [0.5-0.7], yet with a well defined mean $h = 0.57$ [Hack, 1957].

The distributions of (computed) drainage discharge, drainage area and upstream length are plotted in Figure 3.10. In Figure 3.10a, the flume discharge can be separated into low ($q \leq 1.1 \times 10^4 \text{ mm h}^{-1}$), medium ($1.1 \times 10^4 < q < 3 \times 10^6 \text{ mm h}^{-1}$) and high ($q \geq 3 \times 10^6 \text{ mm h}^{-1}$) sections. The low discharge region mostly covers the left of the flume (Figure 3.6) where the precipitation rate is lower. The values of $P(Q > q)$ for these regions do not change during the network evolution (from 0.25 h to 16 h). For the medium discharge regions, a power law relationship ($P(Q > q) = q^{-\varphi}$) describes the exceedance probability with an exponent of $\varphi = 0.49$. The high discharge area shows the most temporal variability, which corresponds to the change of the main streams (A-D in Figure 3.6). Since the D8 algorithm selects a single adjacent down-gradient cell to receive water from a given cell, potentially the predicted flow becomes more localized than in reality. Also, flow disturbances due to raindrop impact and resulting mixing are not accounted for. Due to spatial and temporal variations of precipitation in natural settings, the distributions of drainage area and upstream length are more commonly used metrics for describing river networks at large (spatial) scales. Even though in this study no rills formed, the distributions of drainage area and upstream length under this shallow, overland flow cross a number of scales characterized by power laws ($P(A > a) = a^{-\beta}$ and $P(L > l) = l^{-\psi}$) with

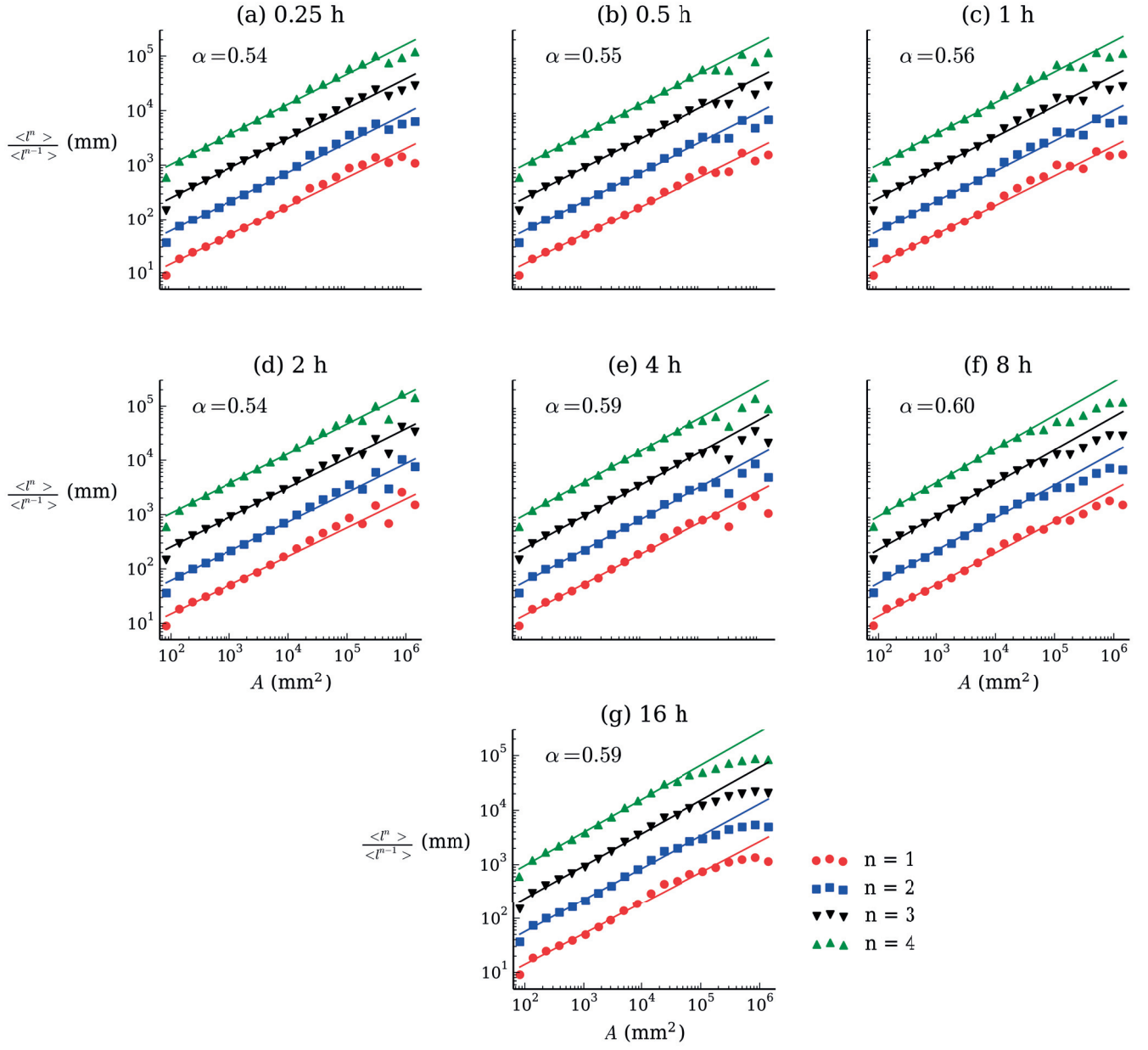


Figure 3.9 – Relation between the upstream length and drainage area ($l = A^h$). The curves of higher moments ($n > 1$) are shifted vertically for the purpose of visualization.

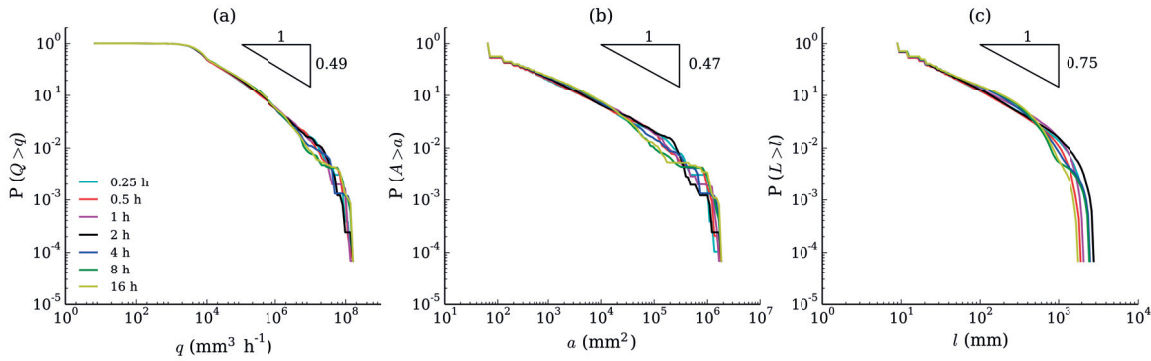


Figure 3.10 – Exceedance probabilities of discharge (Q), drainage area (A) and upstream length (l) at different times.

$\beta = 0.47$ and $\psi = 0.75$, respectively (Figure 3.10b and c). Furthermore, at 10% slope with an average rainfall of 60 mm h^{-1} , exponents of 0.49, 0.47 and 0.71 were found for power laws describing discharge, drainage area and upstream length distributions, respectively (Figure 3.17). These results are very similar to large scale river networks [Mandelbrot, 1982; Tarboton *et al.*, 1989; Rigon *et al.*, 1996; Dodds and Rothman, 2001a, b, c; Rinaldo *et al.*, 2014]. These exponents are close to analytical results, $\beta = 1 - h$ and $\psi = \beta/h$, derived by Maritan *et al.* [1996].

3.3.3 Further morphological features

The scaling relation between the slope and drainage area is plotted in Figure 3.11. Similarly to large scale river networks and channeled surfaces at the laboratory scale [e.g., Sweeney *et al.*, 2015], a power law describes the intermediate drainage areas of the overland flow (Figure 5). Figure 3.11 shows the curvature versus the product of drainage slope and area (slope-area). Similar to the catchment scale [Perron *et al.*, 2009], the average values of curvature are negative where the product of area and slope ($A|\nabla z|$) is low whereas positive curvature values are observed for increasing $A|\nabla z|$ (Figure 3.12).

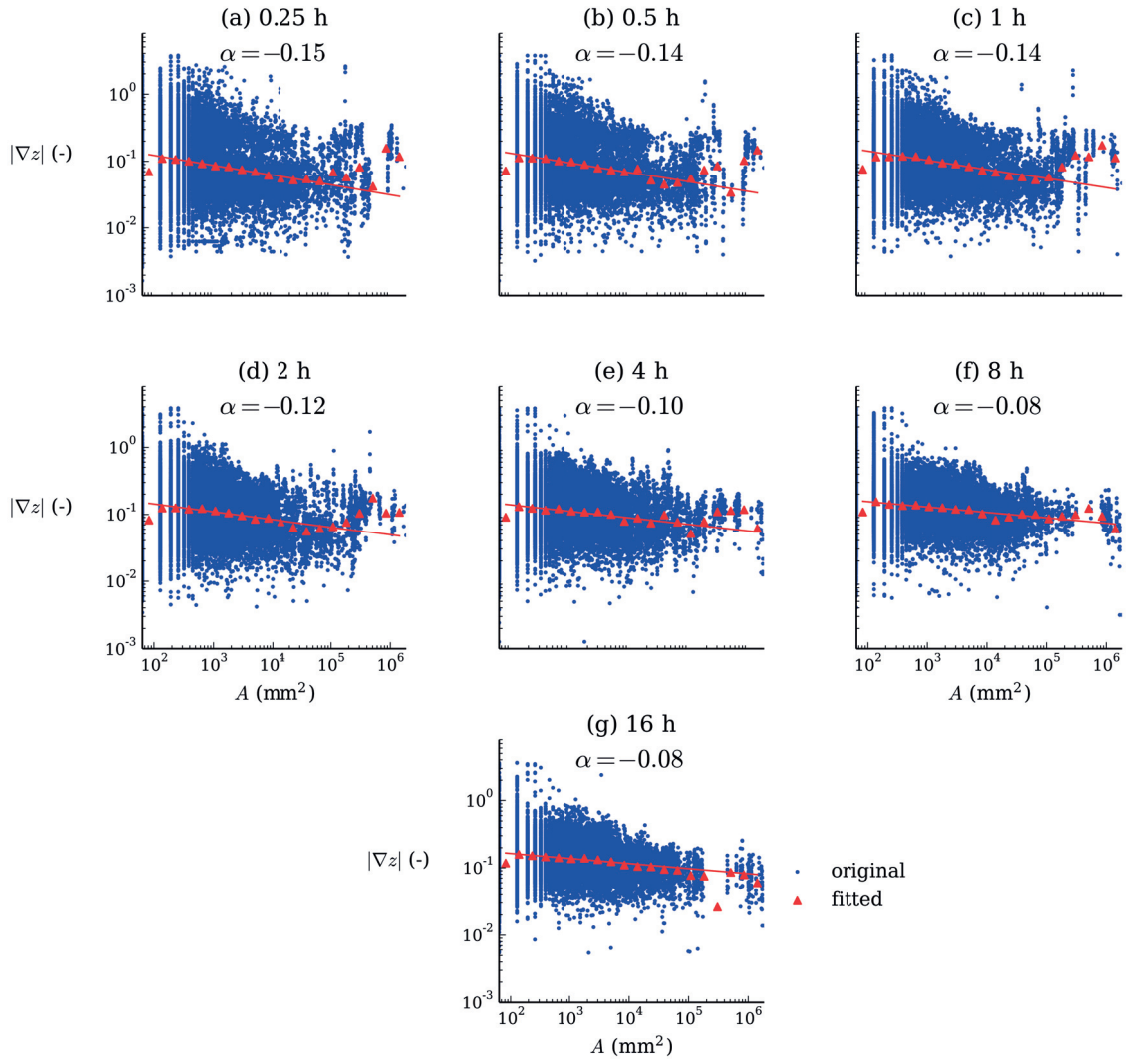


Figure 3.11 – Slope versus drainage area. A power law describes the intermediate drainage areas.

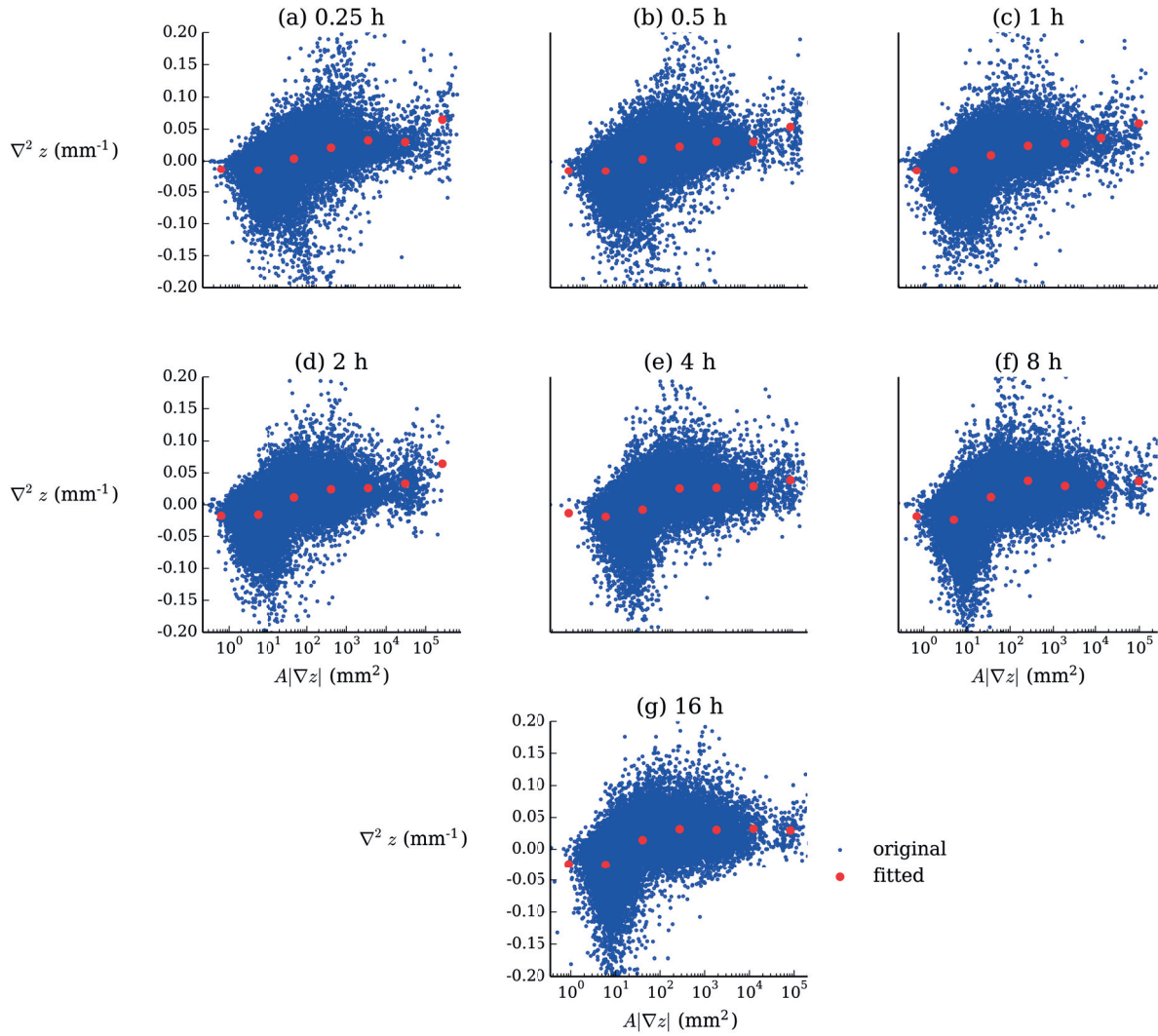


Figure 3.12 – Curvature versus the product of drainage area and slope. The average values of curvature are negative where the product of area and slope ($A|\nabla z|$) is low whereas positive curvature values are observed as $A|\nabla z|$ increases.

3.3.4 Another experiment

In order to investigate the scaling laws at different slope and rainfall intensity, another experiment was carried out at 10% slope and average rainfall of 60 mm h^{-1} for a duration of 20 h. Again, the surface morphology remained unchanneled during whole experiment. The corresponding elevation field and the extracted networks are shown in Figures 3.13-3.15. As shown in Figure 3.16, Hack's law was observed where the exponents were in the range of [0.51-0.55] during the experiment. The distributions of discharge, drainage area and upstream length follow the power laws: $P(Q > q) = q^{-\varphi}$, $P(A > a) = a^{-\beta}$ and $P(L > l) = l^{-\psi}$, with $\varphi = 0.49$, $\beta = 0.47$ and $\psi = 0.71$, respectively (Figure 3.17). Furthermore, the relation between the exponents are close to analytical results, $\beta = 1 - h$ and $\psi = \beta/h$, derived by *Maritan et al.* [1996]. These results show that irrespective of rainfall intensity and slope, the same scaling laws were generated.

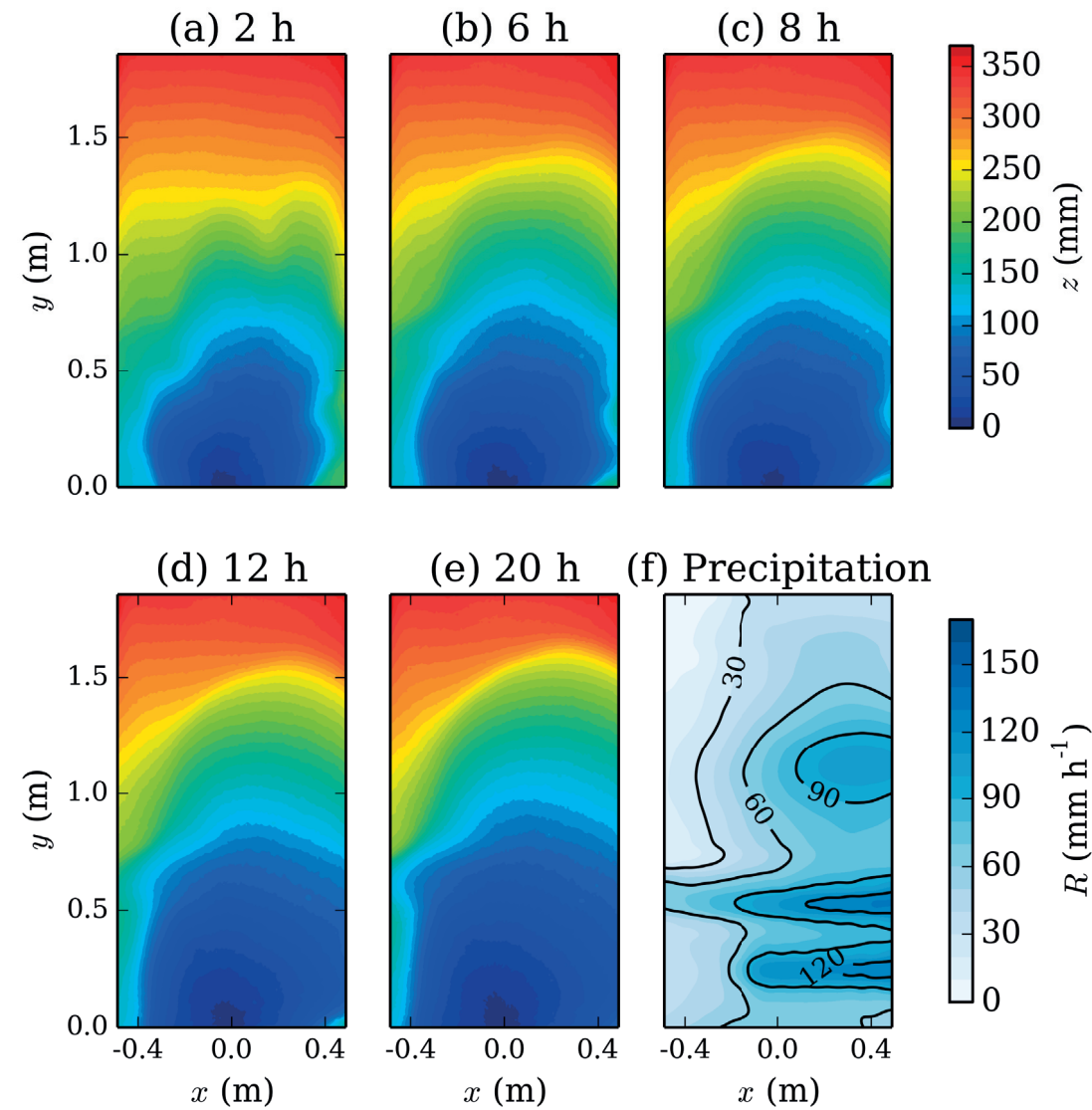


Figure 3.13 – Elevation field at 10% slope and 60 mm h^{-1} average rainfall.

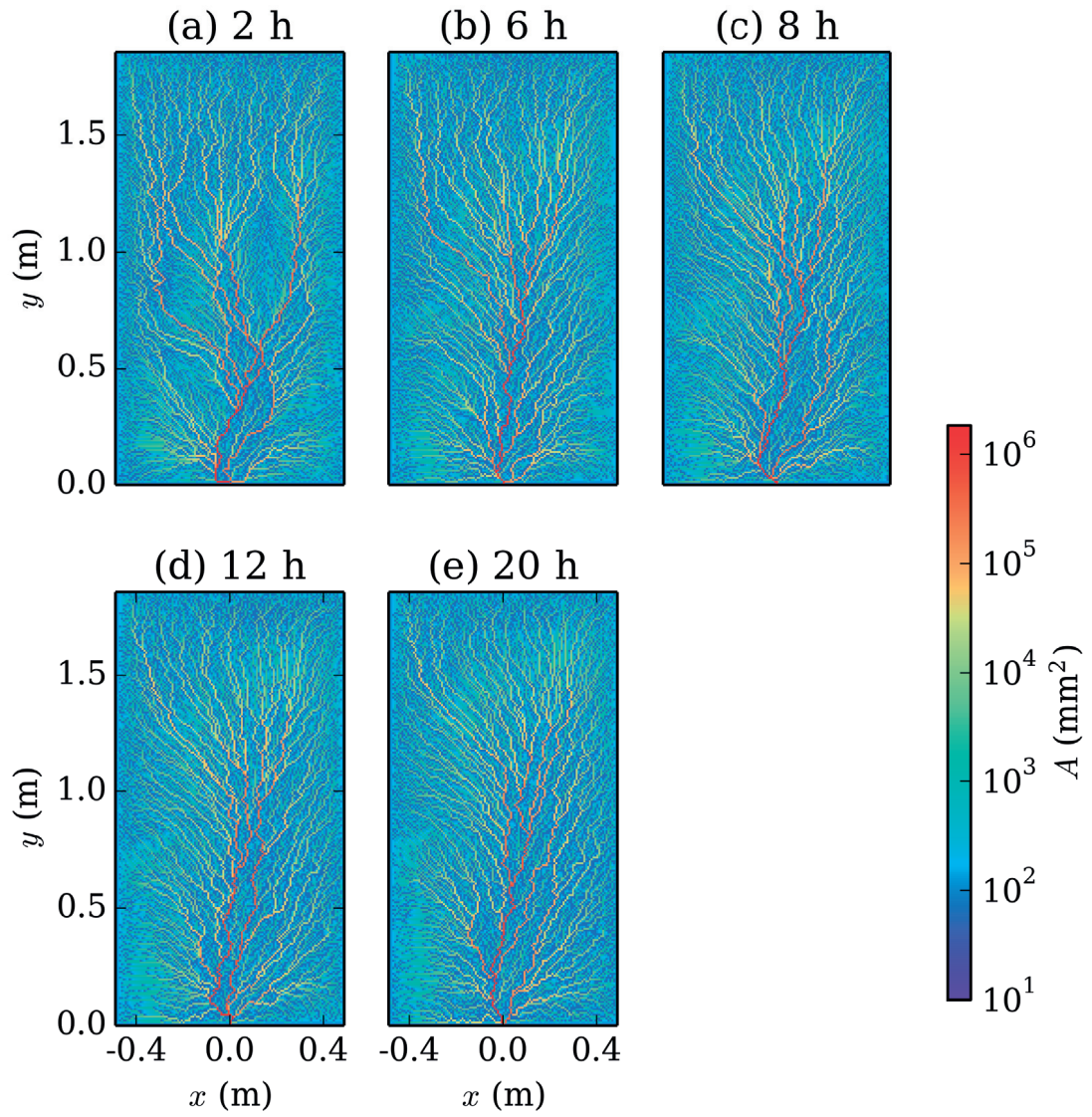


Figure 3.14 – Drainage area at 10% slope and 60 mm h^{-1} average rainfall.

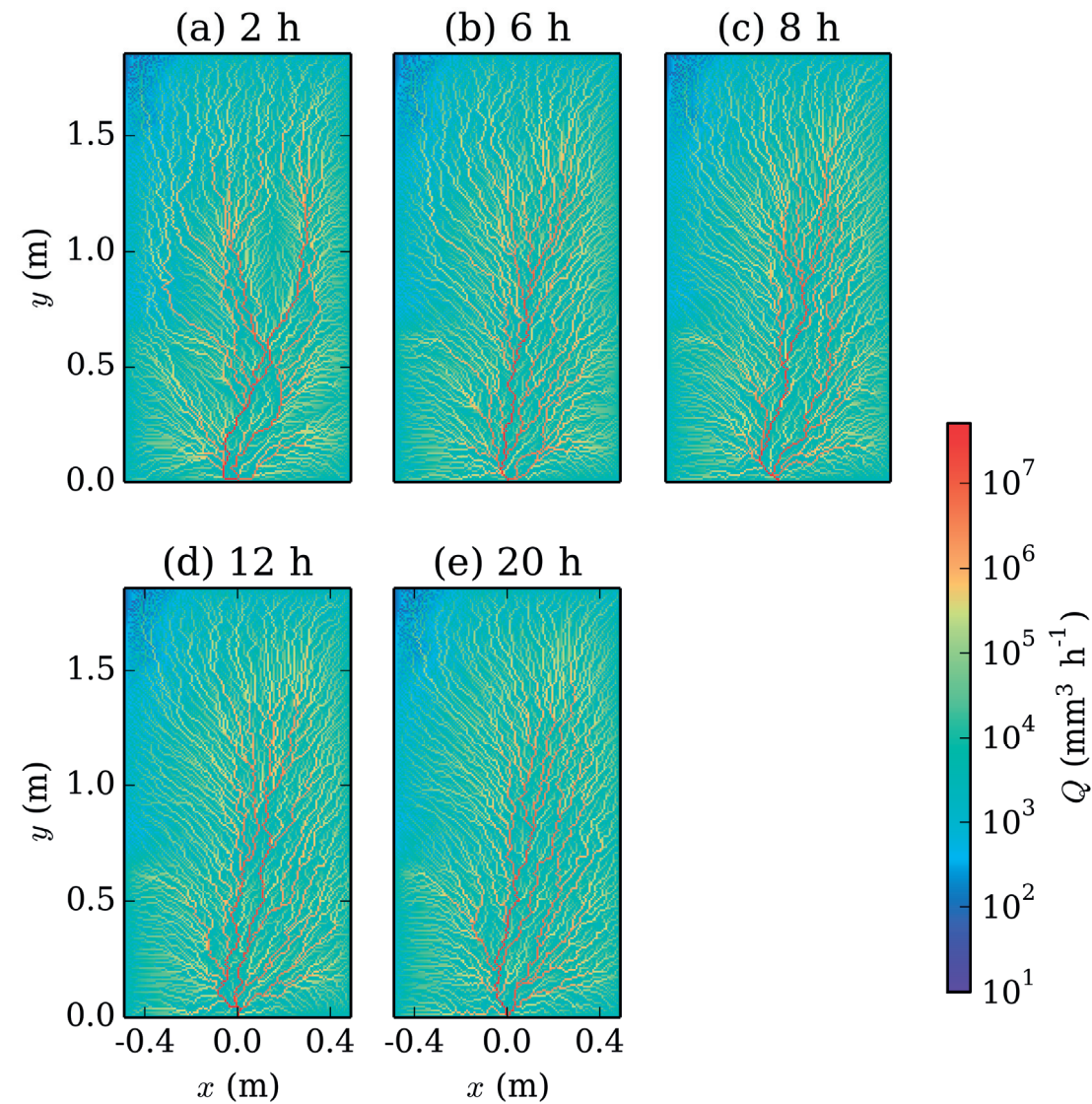


Figure 3.15 – Discharge flux at 10% slope and 60 mm h^{-1} average rainfall.

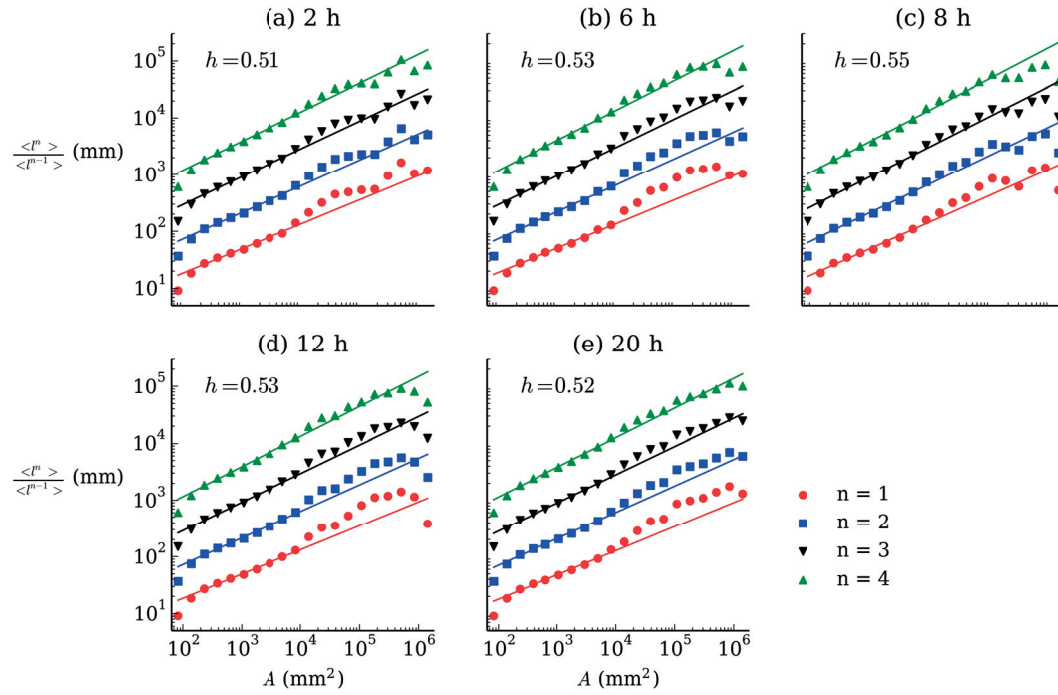


Figure 3.16 – Plot of Hack's law ($l = A^\alpha$) at 10% slope and 60 mm h^{-1} average rainfall. The curves of higher moments ($n > 1$) are shifted for the purpose of visualization.

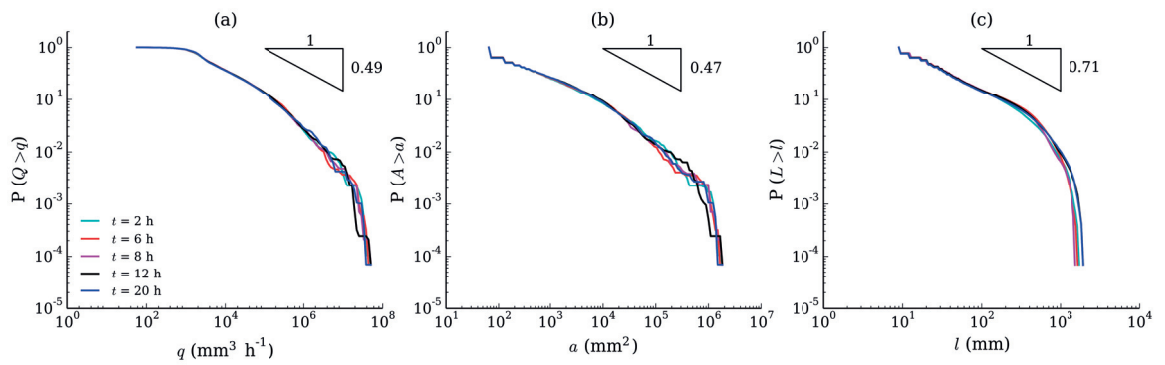


Figure 3.17 – Scaling relation for the distribution of discharge (a), drainage area (b) and length (c) at 10% slope and 60 mm h^{-1} average rainfall.

3.4 Physical interpretation of the invariant scaling laws

The consistency between the laboratory results (Figures 3.9, 3.10, 3.16 and 3.17) and results for catchment networks [e.g., *Rodríguez-Iturbe and Rinaldo, 1997*] points to an underlying governing principle operating at different scales, such as the principle of minimum energy expenditure [*Rodríguez-Iturbe et al., 1992*] that applies at equilibrium conditions for river networks. Similarly, recent work [*Smith, 2018*] on equilibrium landscapes showed that overland flows minimized a Lagrangian function of kinetic and potential energies. For both potential (viscosity dominated) and inviscid flows and for fixed boundary conditions, energy dissipation continues monotonically until the steady flow configuration is achieved, i.e., energy dissipation is a minimum [*Lord Rayleigh, 1893*]. The energy minimization principle has been shown exactly (by re-parametrization invariance arguments, and in the small gradient approximation) to correspond to the steady-state solution of the general landscape evolution equation in fluvial regions [*Banavar et al., 2001*]. Deriving scaling properties and self-organization in optimal networks is therefore tantamount to analyzing the underlying equations if steady-state solutions are sought. Laboratory-scale rill networks were also shown to evolve towards the minimum energy expenditure [e.g., *Gómez et al., 2003; Berger et al., 2010*]. However, for unchanneled morphologies, further investigation is needed since our results suggest (approximately) time-invariant scaling laws for a rapidly eroding surface.

The dynamics of eroding surfaces and related overland flow (including raindrop impact) can be modeled via different approaches, from mechanistic models that consider coupled overland flow and soil erosion [e.g., *Nearing et al., 1989; Hairsine and Rose, 1992a, b*] to catchment scale landscape evolution models (LEMs) [e.g., *Willgoose, 1989; Howard, 1994; Perron et al., 2008; Smith, 2018*]. LEMs, which predict channel networks at both the catchment and laboratory scales, are relevant to our experimental results. We emphasize that our experiment involves continuous overland flow on an unchanneled surface in contrast to channelized flow in a catchment. Nonetheless, characterization of the overland flow on the measured morphology via the D8 algorithm results in a network that is geometrically similar to a catchment drainage network. The D8 algorithm provides a network representation of the overland flow driven by gravity. This representation is an approximation, but allows for a direct comparison of the unchanneled surface morphology in our experiments with the channeled networks found in catchments and in laboratory

experiments.

These experiments support a notable extension of what was previously thought about the kind of recursive features shown by channeled landscapes at much larger scales. Unchanneled landscapes were thought to obey diffusive evolution. For splash-dominated erosion studied here, the scaling structures were replicas of those occurring at orders of magnitude larger scales. It is totally remarkable that the aggregation patterns are independent of the specific sediment transport type in erosional patterns. Moreover, the temporal stability of the scaling structures we measure here suggests that indeed the planar features of steady states are reached almost immediately by erosional surfaces, as was speculated but never shown for real river networks. We suggest that the results could provide a test case for LEMs, which are applicable at both the laboratory [Sweeney *et al.*, 2015] and catchment scales [Perron *et al.*, 2009] on the condition that channels are formed. In the above-mentioned network analysis of Banavar *et al.* [2001], diffusion was ignored, although it is present in LEMs. Since diffusion effects will tend to smooth surfaces in LEM predictions, we speculate that our results will prompt additional investigations of the role of diffusion in these models. That is, it remains to be determined if the scale invariance uncovered in this work can be captured by LEMs.

3.5 Conclusions

An evolving unchanneled surface under a spatially non-uniform rainfall was statistically characterized in the same manner as large scale river networks by converting the continuous overland flow into drainage area and discharge networks. The measurements show that although the surface morphology and the corresponding overland flow network changed markedly during the experiment, the system preserved Hack's law and power laws in distributions of drainage area, length and discharge. More importantly, the exponents, the values of which are identical to large scale river networks, remained in a narrow range despite the considerable change in the surface morphology and the corresponding network structure. This work provides, for the first time, experimental support for the self-similar organization of landscapes even where observable rills or channels are not formed on the surface.

4 Landscape evolution model in absence of rills

Mohsen Cheraghi¹, Andrea Rinaldo^{2,3}, Graham C. Sander⁴, Seifeddine Jomaa⁵, Andrea Cimatoribus¹ and D. A. Barry¹

¹Ecological Engineering Laboratory (ECOL), Institute of Environmental Engineering (IIE), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

²Ecohydrology Laboratory (ECHO), Institute of Environmental Engineering (IIE), School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

³Dipartimento di Ingegneria Civile Edile e Ambientale, Università di Padova, Padua, Italy

⁴Department of Civil and Building Engineering, Loughborough University, Loughborough, UK

⁵Department of Aquatic Ecosystem Analysis and Management, Helmholtz Center for Environmental Research—UFZ, Magdeburg, Germany

To be submitted to Journal of Geophysical Research.

4.1 Introduction

The complexity of natural landscapes reflects the numerous factors involved in their formation such as climate [Han *et al.*, 2015; Francipane *et al.*, 2015; Panagos *et al.*, 2017; Ranjbar *et al.*, 2018], chemical and physical properties of the sediments [Massong and Montgomery, 2000; Sklar *et al.*, 2001; Park and Latrubesse, 2015], episodic gully erosion [Pazzaglia *et al.*, 2015], different vegetation types [Istanbulluoglu and Bras, 2005; Jeffery *et al.*, 2014; Corenblit *et al.*, 2015], tectonic effects [Pedrazzini *et al.*, 2016] and the rate of sediment production [Rodriguez-Lloveras *et al.*, 2015; Reusser *et al.*, 2015; Forte *et al.*, 2016; Sadeghi *et al.*, 2017b], all of which are heterogeneous and uncertain. Landscape formation and evolution are not directly measurable in the field due to the time scales involved. Consequently, suitably designed laboratory experiments are of considerable value in that they provide empirical data that are amenable to testing hypotheses of mechanisms underlying observed geomorphological changes [Paola *et al.*, 2009].

Physically based landscape evolution models (LEMs) [Willgoose, 1989; Willgoose *et al.*, 1991, 1992; Howard, 1994] are useful tools to explain the surface geometry in landscapes and laboratory experiments [Densmore *et al.*, 1998; Mudd, 2016; Whipple *et al.*, 2016, 2017; Sinclair, 2017]. In LEMs, the complex fluid-particle interactions within landscapes are described by a governing equation for the surface elevation, with an additional model for surface flow [Chen *et al.*, 2014a]. For catchments, the focus of surface flow is the stream/river drainage network, rather than the overland flow. Typically, flow is modeled in a simplified manner that conserves the volume flux at each cell in the landscape [O'Callaghan and Mark, 1984; Freeman, 1991; Quinn *et al.*, 1991; Costa-Cabral and Burges, 1994; Tarboton, 1997]. Broadly speaking, the relative importance of the advective and diffusive processes described by the LEM controls the landscape geometry produced. That is, considering an initially smooth morphology, localized (channel-forming) landscape incision is favored when advection (or surface shear stress) dominates, whereas more gradually varying landscapes will result when (effective) diffusion dominates.

There are numerous applications LEMs to understand different features of the surface morphology of natural landscapes [Densmore *et al.*, 1998; Yang *et al.*, 2015; Mudd, 2016; Whipple *et al.*, 2016, 2017; Sinclair, 2017; Theodoratos *et al.*, 2018]. For example, Perron *et al.* [2008] derived an expression for the distance between first-order valleys and validated the formula via measurements from five different natural landscapes

[Perron *et al.*, 2009]. Willett *et al.* [2014] showed that drainage basins that share a common hillslope can exchange the drainage area (due to changes in surface morphology) and thereby reorganize their structure. The criterion defining the reorganization direction were found by using the steady state solution of an LEM [Taylor and Leigh, 2013].

Similar to their application to natural landscapes, LEMs were also used in analysis of laboratory experiments. Such experiments permit exploration of different initial conditions, e.g., a smooth land surface composed of uniform and non-cohesive sediment grains, with surface morphology changes induced by rainfall. Advection-dominated setups focus on the evolution of the network structure (i.e., surface incisions) [Hancock and Willgoose, 2002; Bonnet and Crave, 2003; Hasbargen and Paola, 2003; Lague *et al.*, 2003; Bonnet and Crave, 2006; Paola *et al.*, 2009; Bonnet, 2009; Gravelleau *et al.*, 2012; Rohais *et al.*, 2012; Reinhardt and Ellis, 2015; Singh *et al.*, 2015]. Small raindrop sizes minimize the kinetic energy of raindrop impact, leading to surface morphologies that evolve almost exclusively through shear stress of surface flows. For instance, Hasbargen and Paola [2000] set up an experiment in an elliptical basin (98 cm \times 87 cm) subjected to a constant uplift rate. They pointed out that the oscillation of the erosion rate at steady state was the result of knickpoint migration in the domain. The spatial patterns of landslides and knickpoints in a steady-state landscape were measured by Bigi *et al.* [2006]. They found a power-law relation between the number of landslides and drainage area. In recent laboratory experiments, Sweeney *et al.* [2015] used rainfalls of different droplet sizes applied to landscapes with the same initial condition. For the different landscape patterns so-created, they showed that the drainage density decreases with increasing droplet size (i.e., relative increase in diffusion).

There are numerous similar studies that examined rill formation on hillslopes [e.g., Parker, 1977; Bryan and Poesen, 1989; Gomez and Mullen, 1992; Brunton and Bryan, 2000; Römkens *et al.*, 2002; Pelletier, 2003; Raff *et al.*, 2004; Nearing *et al.*, 2004; Tatard *et al.*, 2008; Yao *et al.*, 2008; Oliveto *et al.*, 2010; Gordon *et al.*, 2011, 2012; Stefanon *et al.*, 2012; Shit *et al.*, 2013; He *et al.*, 2014; Bennett *et al.*, 2015; Bennett and Liu, 2015]. In these experiments, droplet sizes are large enough to induce splash-impact erosion, which is manifested in changes in surface morphology. For instance, Gómez *et al.* [2003] and Berger *et al.* [2010] tested the minimal energy expenditure theory [Rinaldo *et al.*, 1992] for rill networks considering different slopes, rainfall intensities and initial conditions. They found that, when the land was effectively incised,

Chapter 4. Landscape evolution model in absence of rills

the total energy dissipation decreased as the rills evolved. *McGuire et al.* [2013] calibrated experimental data with the shallow water flow equations coupled with a process-based erosion model [*Hairsine and Rose*, 1991, 1992a, b; *Simpson and Schlunegger*, 2003; *Simpson and Castelltort*, 2006]. They found that the deposition term in the equations had a minor effect on the (qualitative) match between the model and experiment.

Highly heterogeneous factors of landscapes make their evolution more complex than purely random or deterministic physical models. *Hasbargen and Paola* [2000] showed that even at the ideal condition of laboratories, the chaotic feature of morphology evolutions is unavoidable. In this regard, landscape evolution is claimed to be comparable with turbulence in fluid mechanics [*Passalacqua et al.*, 2006]. The comparison is not surprising as the surface morphology is extremely coupled with the turbulent flow on the landscape.

Stochastic Partial Differential Equations (PDEs) are known to be effective approaches that can represent the complexity of processes, heterogeneity of properties and inevitable uncertainties, along with simplicity in calibration and parameter estimation [*Foufoula-Georgiou et al.*, 2010; *Fatichi et al.*, 2016]. SPDEs were previously introduced as landscape evolution models [e.g., *Giacometti et al.*, 1995; *Caldarelli et al.*, 1997; *Pastor-Satorras and Rothman*, 1998; *Pelletier*, 2007; *Bonetti and Porporato*, 2017]. For instance, *Giacometti et al.* [1995] introduced a fourth order PDE (inspired by *Edwards and Wilkinson* [1982]) with a random noise to create a self-organized system. However, the models were not calibrated with experimental or field landforms to see how efficiently the models could reproduce the elevation field and its corresponding scaling laws. In addition to using stochastic PDEs, underlying theories and their interpretations developed to simulate more realistic landscape evolution patterns [e.g., *Gabet and Mendoza*, 2012; *Furbish and Roering*, 2013; *Benson et al.*, 2013; *Chen et al.*, 2014b; *Goren et al.*, 2014; *Ancey et al.*, 2015; *Houssais et al.*, 2015; *Dorrell et al.*, 2018; *Baumgarten and Kamrin*, 2018; *Calvert et al.*, 2018]. For example, a non-local hillslope evolution was proposed by *Foufoula-Georgiou and Stark* [2010] in which they considered a model with fractional derivatives. Through one-dimensional analysis, they found agreement between the hillslope profile from this model and three field sites. *Ferdowsi et al.* [2018] proposed a glassy sediment flux model for hillslopes which could mimic both slow creeping and fast landslides.

In Chapter 3, we reported experiments with two significant differences to those described above. First, the rainfall was non-uniform and, second, the surface

morphologies were not incised (i.e., rills did not form). It was shown that the morphology keeps the invariant catchment-scale power laws during its evolution. In other words, the statistical features of fluvial parts were found in the unchanneled area. In this Chapter, we consider the unchanneled morphology as an advective-diffusive domain by testing the applicability of a widely-used LEM [Howard, 1994; Perron *et al.*, 2008] for unchanneled morphology evolution. The LEM is described in section 1.3.3. Previous applications of the LEM to catchment scales produced an incised landscape in which discharge occurs either as overland flow (to channels/rivers), or as flow within the channels. In our experiment, only overland flow occurred, and the surface was never incised. Nevertheless, we model the overland flow as a discharge network, as done in other applications of the LEM. Additionally, we test the ability of the model to simulate directly the measured surface morphology, a test that is not possible at the landscape scale, and was hitherto not attempted at the laboratory experiment scale. After analyzing the LEM with uniform coefficients, we take advantage of our high-resolution experimental data to test some model modification scenarios. Naming the original LEM as Model 1, we investigate seven different model modifications (Models 2-8) in three general methodologies: I) Using a spatially non-uniform diffusion coefficient correlated with the rainfall distribution (Model 2), II) Solving the LEM as stochastic PDEs (Models 3-6) and III) A deterministic method that assumes a critical curvature for the onset of the diffusion process (Model 8). In the stochastic form of the LEM, we separately applied spatial stochastic values for the diffusion coefficient (Model 3), the advective coefficient (Model 4), and the critical stream power (Model 5). Furthermore, we added a stochastic noise term to the model which was applied independent of diffusion coefficient (Model 6) and coupled with it (Model 7). In all cases, the parameters were calibrated using the experimental data via a multi-objective optimization method (Borg MOEA [Hadka and Reed, 2013]). The goal was to find out the best scenarios that have the minimum difference with the experimental morphology and preserve of the scaling relations during evolution.

4.2 Numerical simulation and calibration

The sediment morphology after 15 minutes rainfall was used as the initial condition for numerical simulation of the model (equation 1.16). At each time step, the pit

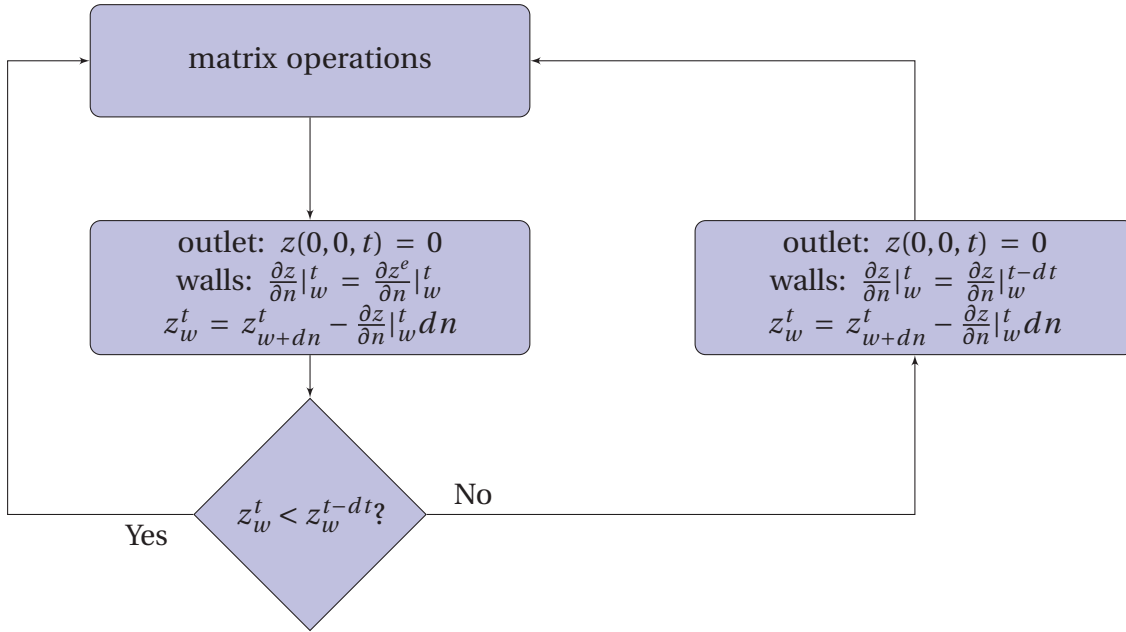


Figure 4.1 – The wall boundary condition assuming \vec{n} as the unit vector normal to walls in outward direction and dn is the discretization size along \vec{n} . $\frac{\partial z^e}{\partial n}|_w^t$ was calculated by linear interpolation between the two consecutive scans before and after the time t .

points were removed using a fast algorithm introduced by *Planchon and Darboux* [2002]. After determining the flow direction via the D8 algorithm [*O'Callaghan and Mark*, 1984], the discharge was calculated by:

$$Q_i = \sum_{j=1}^8 w_{ji} Q_j + R_i dx dy \quad (4.1)$$

where j represents the 8 cells around the i th cell. The value of w_{ji} (–) is one if the cell j flows into cell i , otherwise it is zero. R_i (mm h^{-1}) is rainfall intensity at the cell i (Figure 3.4h) and dx (mm) and dy (mm) are the grid sizes in x and y directions. Because of very long simulation and optimization time, the cell sizes of 8 mm by 8 mm were used for discretization. The D8 algorithm was used to extract the discharge network from the morphology as it is the most common approach at large scales [*Ersine et al.*, 2006]. The details of water velocity field will be lost by approximating the continuous overland flow as a network. However, the algorithm provides a simple approach to approximate the driving shear forces on the morphology.

The numerical solution was obtained by two fractional time steps [*Press et al.*, 2007]. The first one was the second order Runge-Kutta scheme to solve the advection term

(equation 1.16) and the second was implicit ADI algorithm for the diffusion part. More details of the numerical formulation and discretization are explained by *Perron et al.* [2008]. To overcome the wall effects, a conditional boundary condition was used as shown in Figure 4.1. At each time step, after the matrix operations (Runge-Kutta + ADI), the gradient normal to the boundary ($\frac{\partial z}{\partial n}|_w^t$) was calculated by using the experimental data ($\frac{\partial z^e}{\partial n}|_w^t$) which was a temporally linear interpolation of the scans. Afterwards, the sediment elevations adjacent to the walls (z_w^t) were calculated based on the gradients. In the decision block, the value of (z_w^t) was checked; the condition was that the elevation had to decrease at each time step ($z_w^t < z_w^{t-dt}$). If the condition was not met, the slope would remain unchanged (equal to the last time step ($t - dt$)) and based on that, the new elevation at the boundary was calculated.

The parameters D , K and Ω_{cr} in equation 1.16 were calibrated by using the experimental results. As mentioned in section 3.2, the sediment morphology was captured at 0.25 (initial condition), 0.5, 1, 2, 4, 8 and 16 h. The objective function was defined as the mean square error between the model and experiment at 8 h. Particle Swarm Optimization method [*Kennedy*, 2010] was used for the calibration. An in-house C++ code was written for the numerical modeling and optimization processes. The main scripts are listed in Appendices C and D. The corresponding files are available at: <https://github.com/mcheraghi/Landscape-Evolution-Model.git>.

4.3 Model parameter study

The numerical model was verified via analytical results of a 1-D advection-diffusion equation and a 2-D Poisson equation (Appendix B). The model was further tested by changing D in the model (equation 1.16) while $K = 0.2 \text{ mm}^{-\frac{1}{2}} \text{ h}^{-\frac{1}{2}}$, $\Omega_{cr} = 10 \text{ mm}^{\frac{3}{2}} \text{ h}^{-\frac{1}{2}}$. The simulations started by the experimental DEM after 0.25 h (Figure 3.4). The parameter D changed from 0 to $10000 \text{ mm}^2 \text{ h}^{-1}$ and the resulted morphologies after 1 h of simulation are plotted in Figure 4.2. As can be seen, with $D = 0$, the surface is completely incised and as the value of D increases, the incision rate on the surface decreases due to more smoothing effect of the diffusion term.

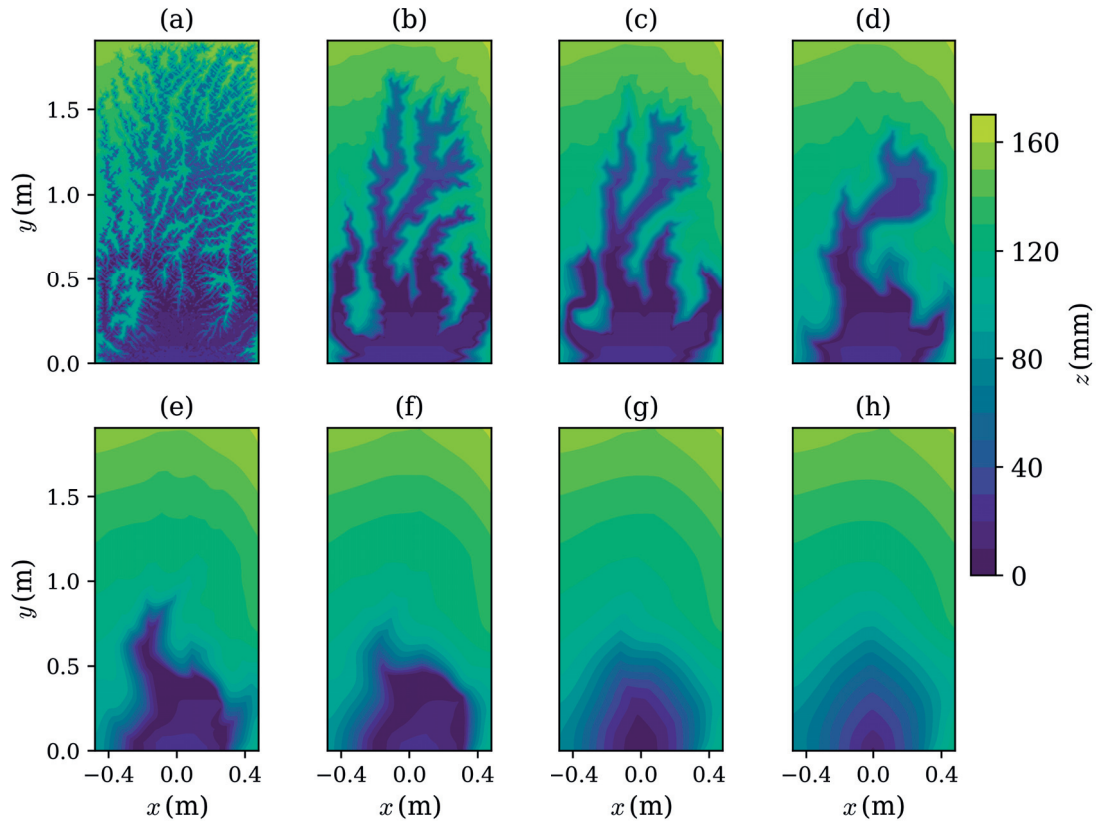


Figure 4.2 – Model results after 1 h for $K = 0.2 \text{ mm}^{-\frac{1}{2}} \text{ h}^{-\frac{1}{2}}$, $\Omega_{cr} = 10 \text{ mm}^{\frac{3}{2}} \text{ h}^{-\frac{1}{2}}$ and D ($\text{mm}^2 \text{ h}^{-1}$) = 0 (a), 250 (b), 500 (c), 1000 (d), 2000 (e), 4000 (f), 8000 (g), 16000 (h). The incision rate of the surface decreases as the diffusion coefficient (D) increases.

4.4 Results of the original model (uniform parameters)

4.4.1 Morphology

Here, we critically examine the LEM’s ability to reproduce the observations of Chapter 3. An obvious advantage of our laboratory experiment is that uncertainties inherent in field-based analyses are avoided. In addition, we use the model to predict the measured morphology directly as well as the statistical feature.

The calibrated parameters are presented in Table 4.1. The diffusion coefficient (D) is higher than those reported for the field scale ($0.16\text{--}222.4\text{ mm}^2\text{ h}^{-1}$) [*Benaïchouche et al.*, 2016; *Martin*, 2000]. This is consistent with the lack of rilling in our experiments, as rill formation is favored for lower diffusivity [*Sweeney et al.*, 2015]. In contrast to the present experimental setup, previous laboratory experiments by which the LEM was analyzed used fine particles, low rainfall intensities and small droplets, i.e., they were designed to produce rills [*Hancock and Willgoose*, 2002; *Bonnet and Crave*, 2003; *Hasbargen and Paola*, 2003; *Lague et al.*, 2003; *Bonnet and Crave*, 2006; *Paola et al.*, 2009; *Bonnet*, 2009; *Graveleau et al.*, 2012; *Rohais et al.*, 2012; *Reinhardt and Ellis*, 2015; *Singh et al.*, 2015; *Sweeney et al.*, 2015].

Table 4.1 – Calibrated parameters for the original LEM (Model 1).

$D\text{ (mm}^2\text{ h}^{-1}\text{)}$	$K\text{ (mm}^{-\frac{1}{2}}\text{ h}^{-\frac{1}{2}}\text{)}$	$\Omega_{cr}\text{ (mm}^{\frac{3}{2}}\text{ h}^{-\frac{1}{2}}\text{)}$
17571	0.184997	13.0765

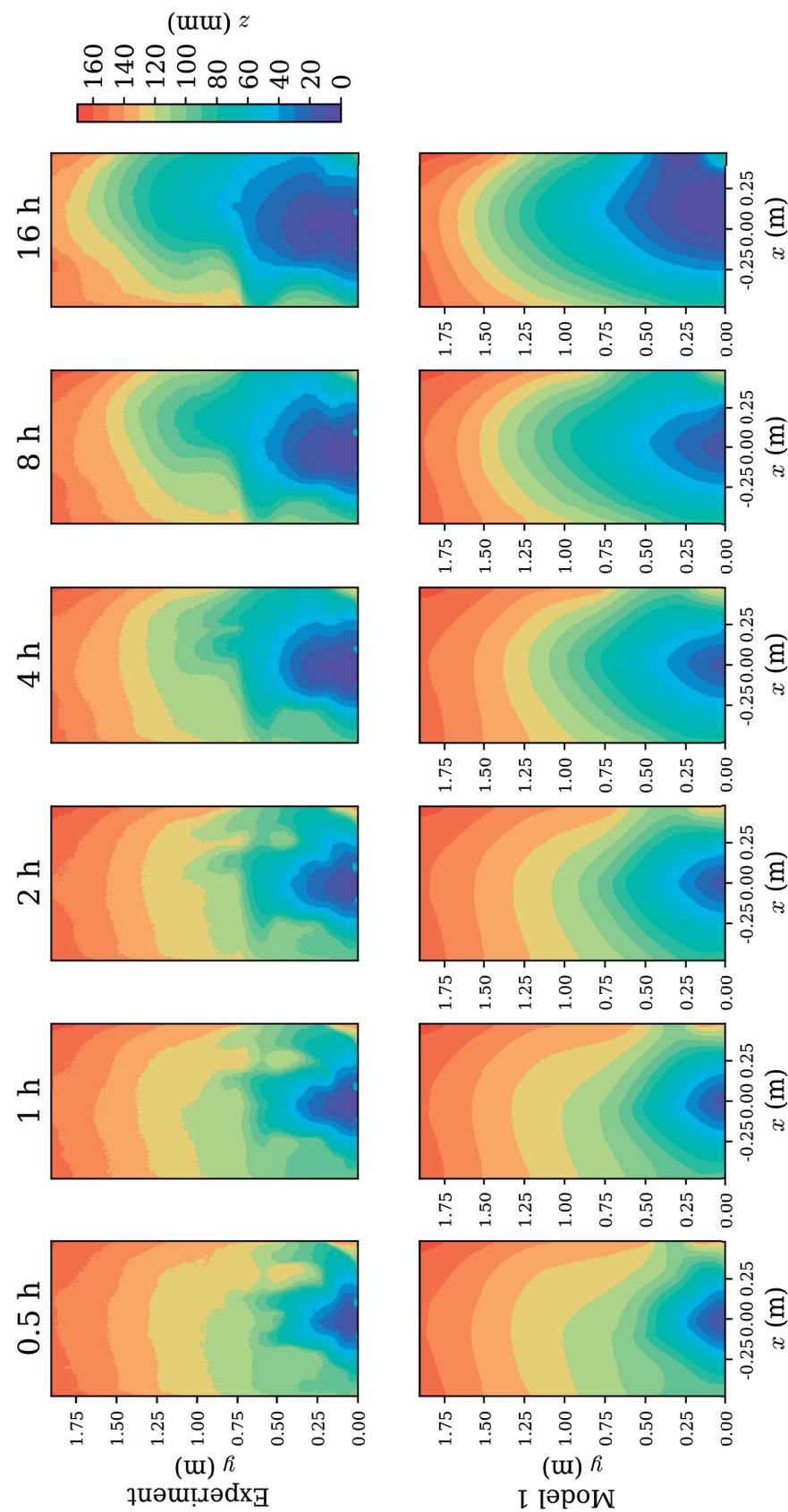


Figure 4.3 – Numerical simulation of the LEM. For comparison, the experimental data (Figure 3.4) are repeated in this figure.

4.4. Results of the original model (uniform parameters)

The numerical results are compared with the experimental data in Figure 4.3. the model is able to capture the main characteristics of the morphology, i.e., the downstream ($z < 60$ mm) have a symmetric shape while the upstream ($z \geq 60$ mm) is being eroded. In agreement with the experiment, the model shows a noticeable movement of downstream at $t = 4$ and 8 h. Recall that the model was calibrated using measurements at 8 h, and then used to predict the morphology at $t = 16$ h, where the agreement is satisfactory.

Despite simulating the main characteristics of the morphology evolution, some differences are seen between the model and the experiment. These differences are likely due to the local scale fluid-particle and particle-particle interactions that are not accounted for in the model. One of these processes is the armoring effect [Polyakov and Nearing, 2003; Wang *et al.*, 2014; Lisle *et al.*, 2017]. Due to shorter erosion time scales, the fine sediment particles are rapidly removed while the larger particles are deposited on the surface or are not moved at all, resulting in a surface covered by pebbles (Figure 3.8). Note that even at low values of the diffusion coefficient, the micro-roughnesses of the hillslopes are always smooth in the model (as we have pure diffusion on the hillslopes due to the step function in equation 1.16). Therefore, lack of agreement between the model predictions and measurements is not related to the parameter values but to some physical processes that are neglected in the model. Nonetheless, the LEM can reproduce with reasonable skill the main features of the morphology by calibrating only three parameters.

4.4.2 Spectral analysis and filtering

The networks in Figure 3.6 were generated by the experimental results where the micro-roughnesses created low order streams. However, because the micro-roughness of the surface is not captured, the LEM is not expected to render a detailed network like the experiment (Figure 3.6). The 2-dimensional Fourier transform of the morphology are shown in Figure 4.4. The zero-wave number component are shifted in the center of domain. The cross in the PSD contours is due to asymmetric boundaries of the elevation domain. The spatially averaged power spectral density (PSD) is plotted in Figure 4.5 as a function of the horizontal wave number (azimuthally averaged). The PSD of the two models diverge from the experimental PSD at a wave number range of $5-9 \text{ m}^{-1}$. In order to remove the details of the discharge networks

Chapter 4. Landscape evolution model in absence of rills

of Figure 3.6 and identify the main streams, the data were low-pass filtered by using the Blackman window [Blackman and Tukey, 1958] with a cutoff wave number of about 7 m^{-1} . In Figure 4.6 the original morphologies of the model and experiment are compared with the filtered ones. The micro-roughness of the experiment (the large deposited stones shown in Figure 3.8b) were removed while the model results are much less sensitive to the smoothing. In Figure 4.7, the discharge was calculated via equation (4.1) using the original and filtered morphologies. After removing the micro-roughness, the detailed network of the experimental results is converted to a network that demonstrates the main flow paths on the flume. When the effect of the large stones are removed, the low order streams of the experimental network are removed and a more continuous field is generated. On the other hand, the discharge network of both models are a little changed after removing the high wave numbers. Therefore, after filtering, extracted network from the experimental and model results show a good agreement.

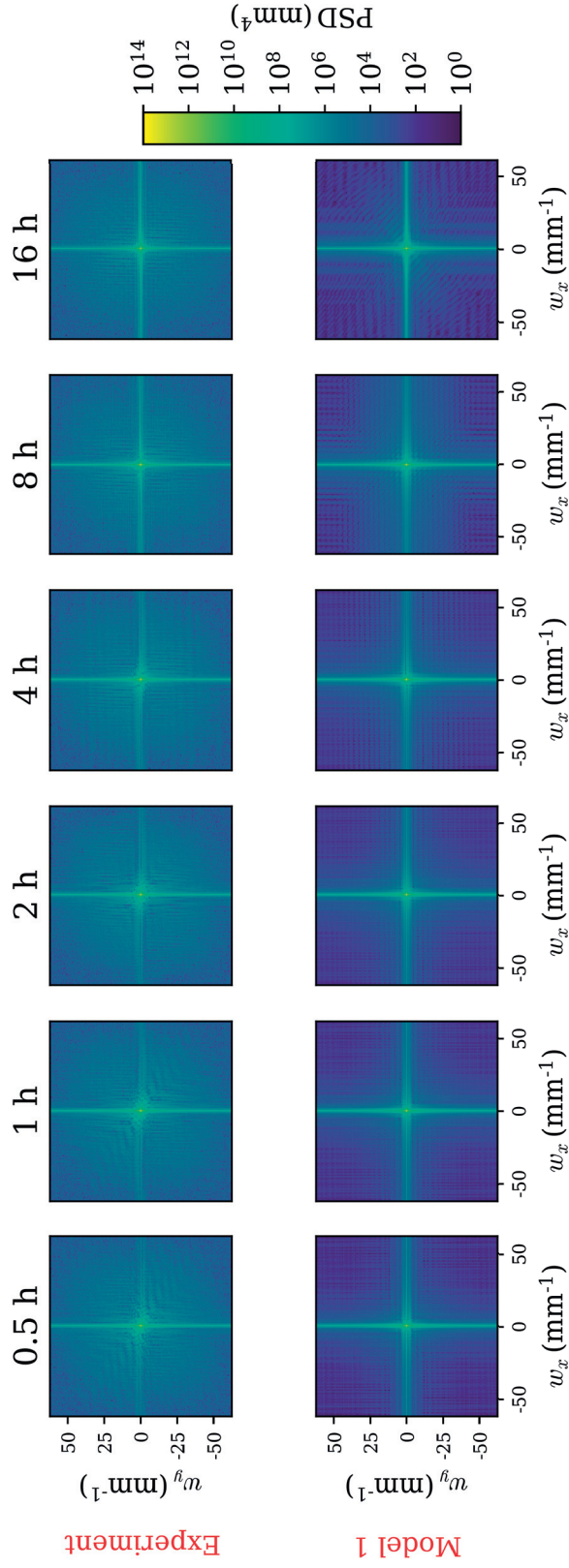


Figure 4.4 – 2-D power spectral density of the experimental and model results. Low wave numbers are more in agreement.

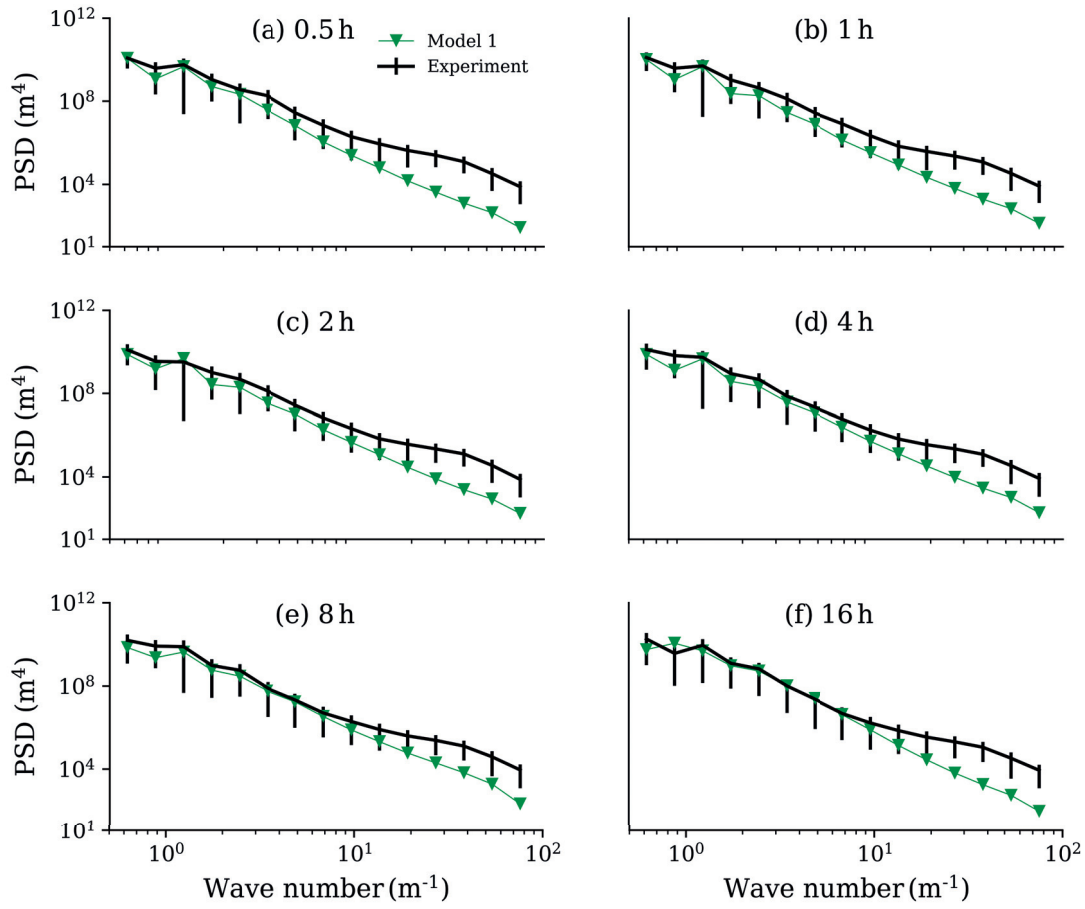


Figure 4.5 – Power spectral density of the morphology. The errorbars of the experimental data are based on the standard deviation from the azimuthally averaged PSD in the wave number domain.

4.4. Results of the original model (uniform parameters)

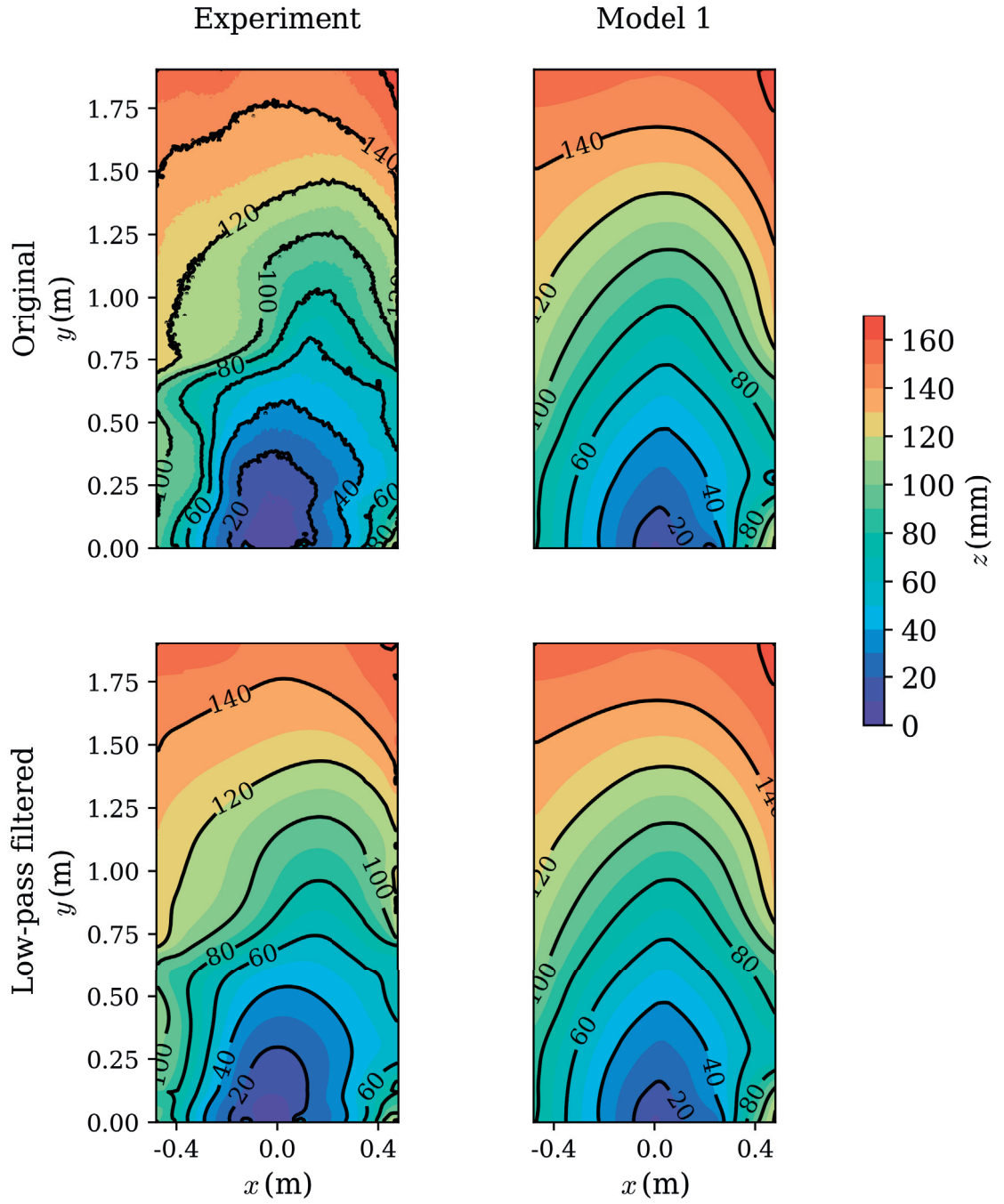


Figure 4.6 – The original and low-pass filtered morphologies at $t = 8$ h.

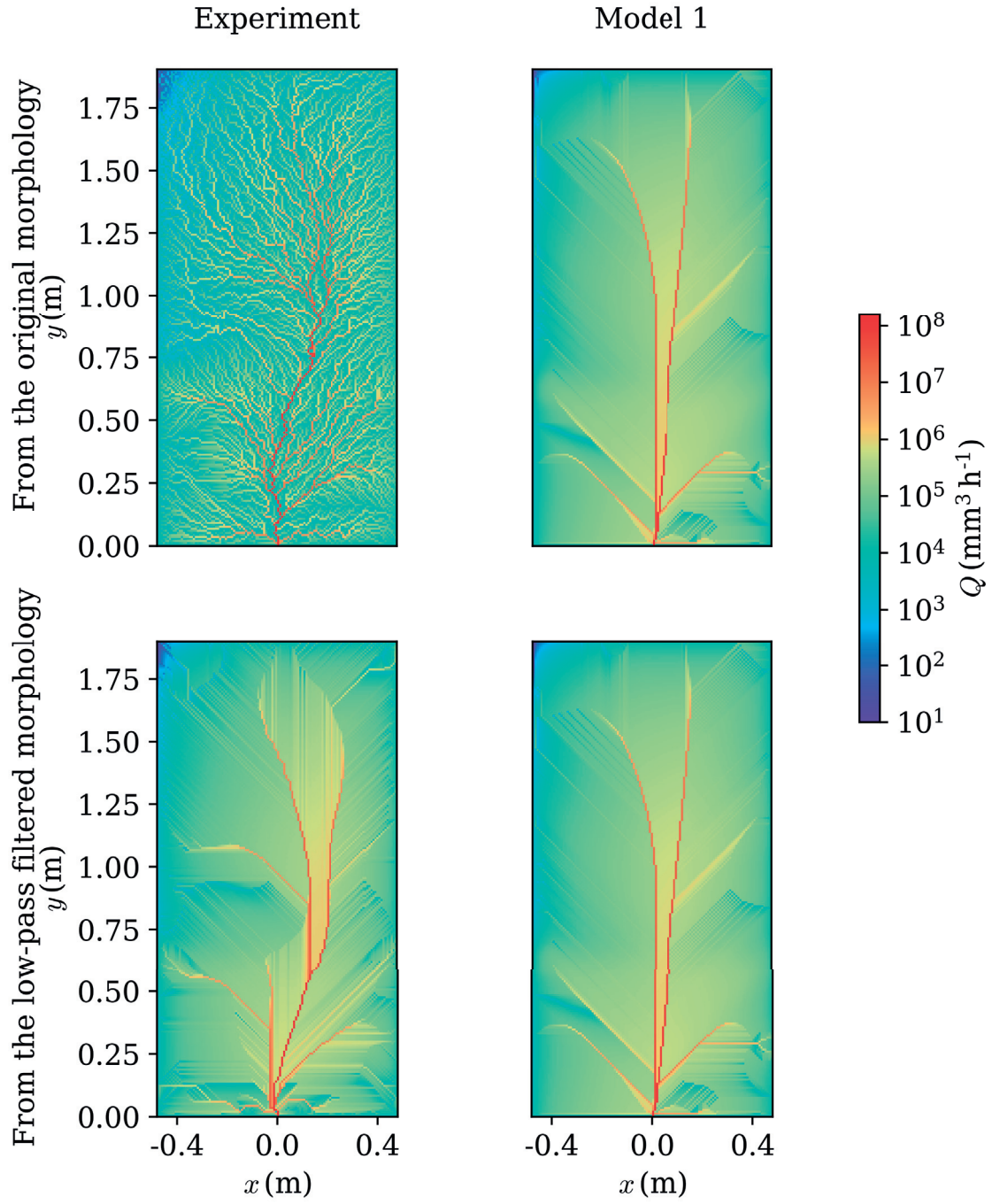


Figure 4.7 – The original discharge network and the network extracted from the low-pass filtered morphology at $t = 8$ h.

4.4.3 Discharge drainage network

The networks in Figure 3.6 were generated directly from the measured morphology, i.e., with small-scale surface roughness included. The small-scale roughness manifests itself in the computed overland patterns. However, the surface roughness is not captured by the LEM, and so the resulting flow patterns are much less detailed than the original experimental data (Figure 3.6). The discharge network of the filtered experimental results are compared with the model network in Figure 4.8. In both experiment and model, the network has a dendritic form at $t = 0.5$ h. Afterwards, the network becomes more concentrated and the flows from the upstream are directed to two main streams. The two streams migrate from left (low precipitation) to the right of the flume (higher precipitation) as the experiment goes on. This migration toward the high precipitation rate area is in agreement with the catchment-scale analysis of *Abed-Elmdoust et al.* [2016] based on optimum channel networks. This migration is responsible for the dynamic change in the erosion pattern of the upstream (Figure 4.3). The model is able to reproduce the migration of the concentrated flow to the right hand side, which induces the morphological evolution of the downstream.

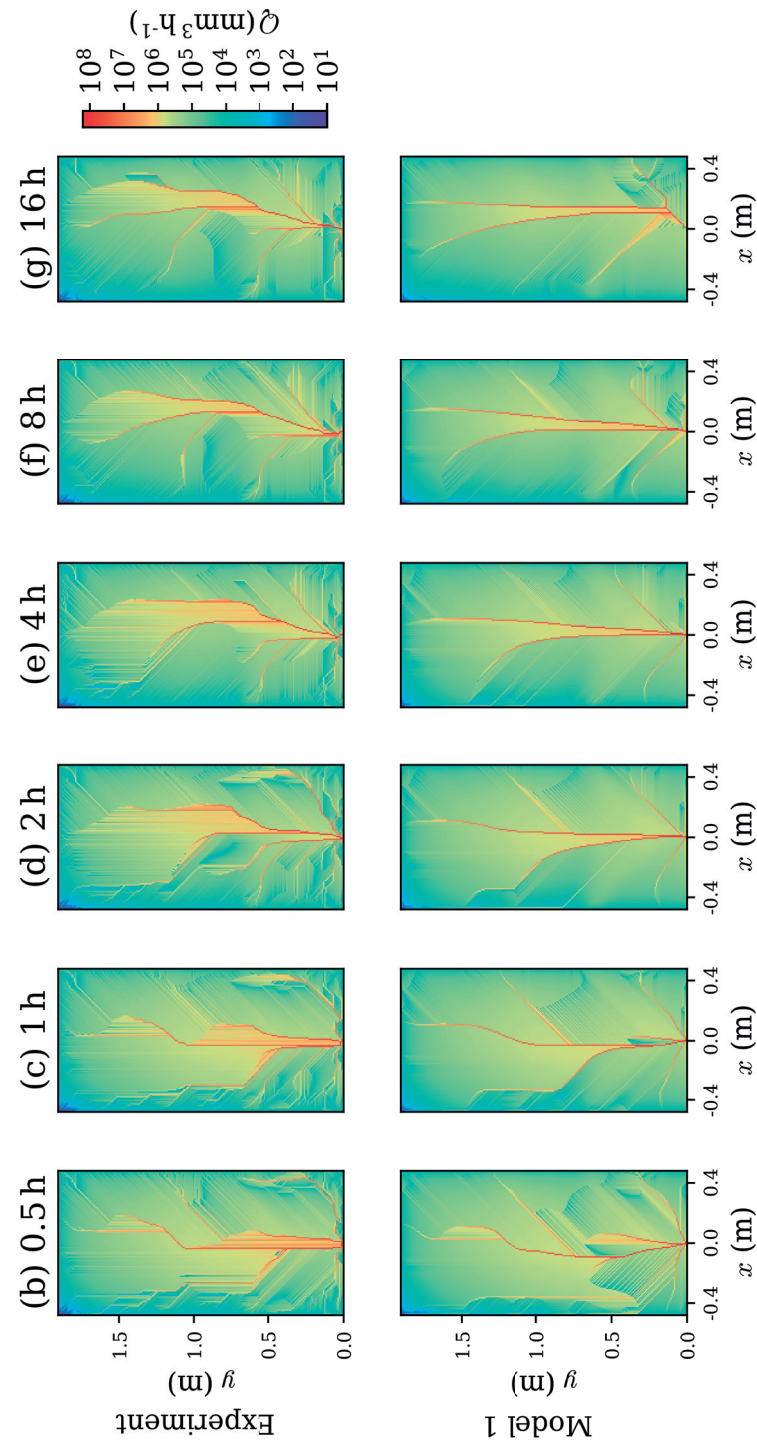


Figure 4.8 – Discharge distribution at different times. The experimental network was calculated based on the filtered morphology.

4.4.4 Statistical analysis

We return to the discharge exceedance probabilities of the flow network generated from the measured morphologies, as shown in Figure 3.10. An obvious question is whether the corresponding discharge network computed for the modeled morphology reproduces that for the measured morphology. The agreement is poor, as shown in Figures 4.9-4.11. This relationship for the exceedance probabilities changed for the filtered morphology to a curve with an inflection point (similar to the model results). Overall, the model can statistically reproduce the filtered experimental data, but not the original data.

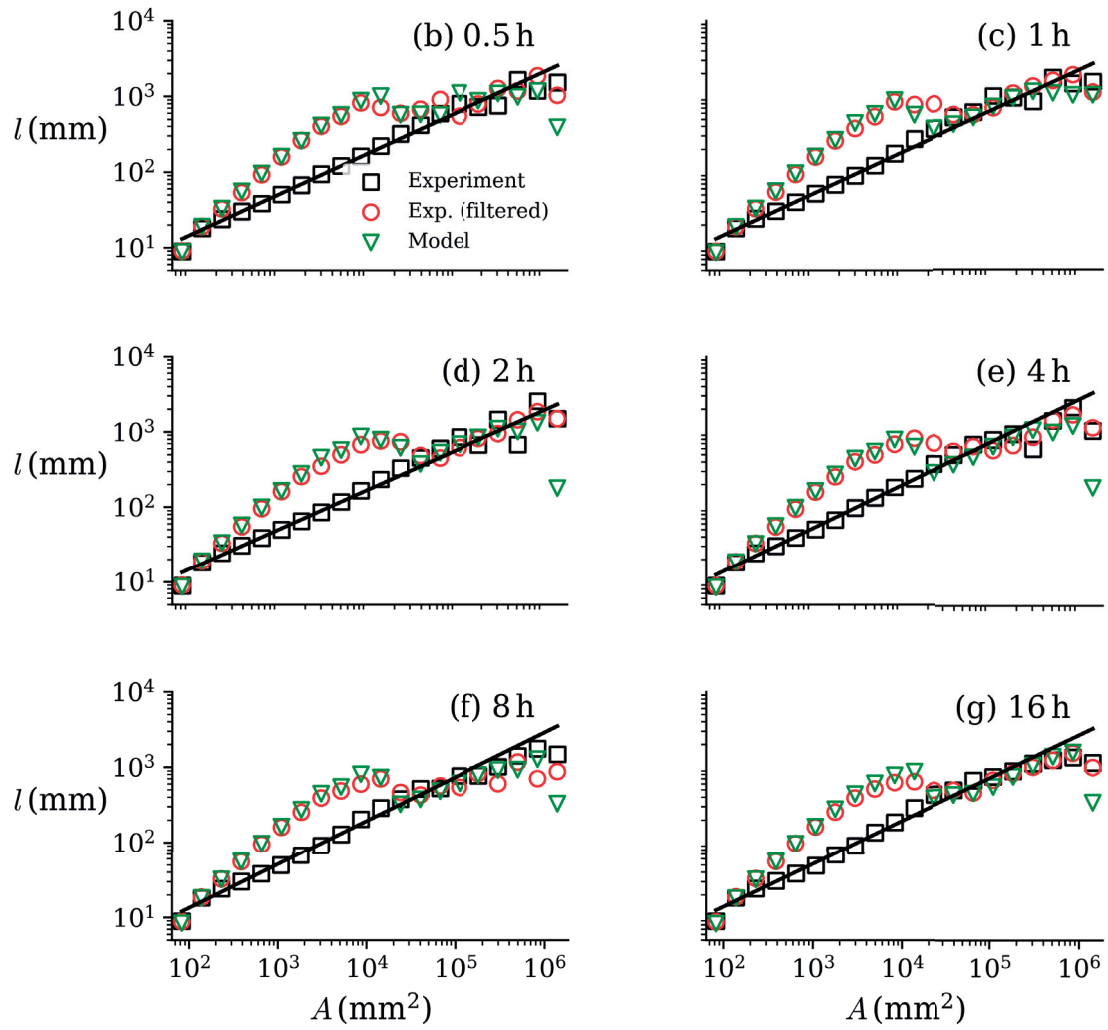


Figure 4.9 – The relation between the upstream length and drainage area (Hack's law) for the model and experiment. The model is able to reproduce the filtered experimental data.

4.4. Results of the original model (uniform parameters)

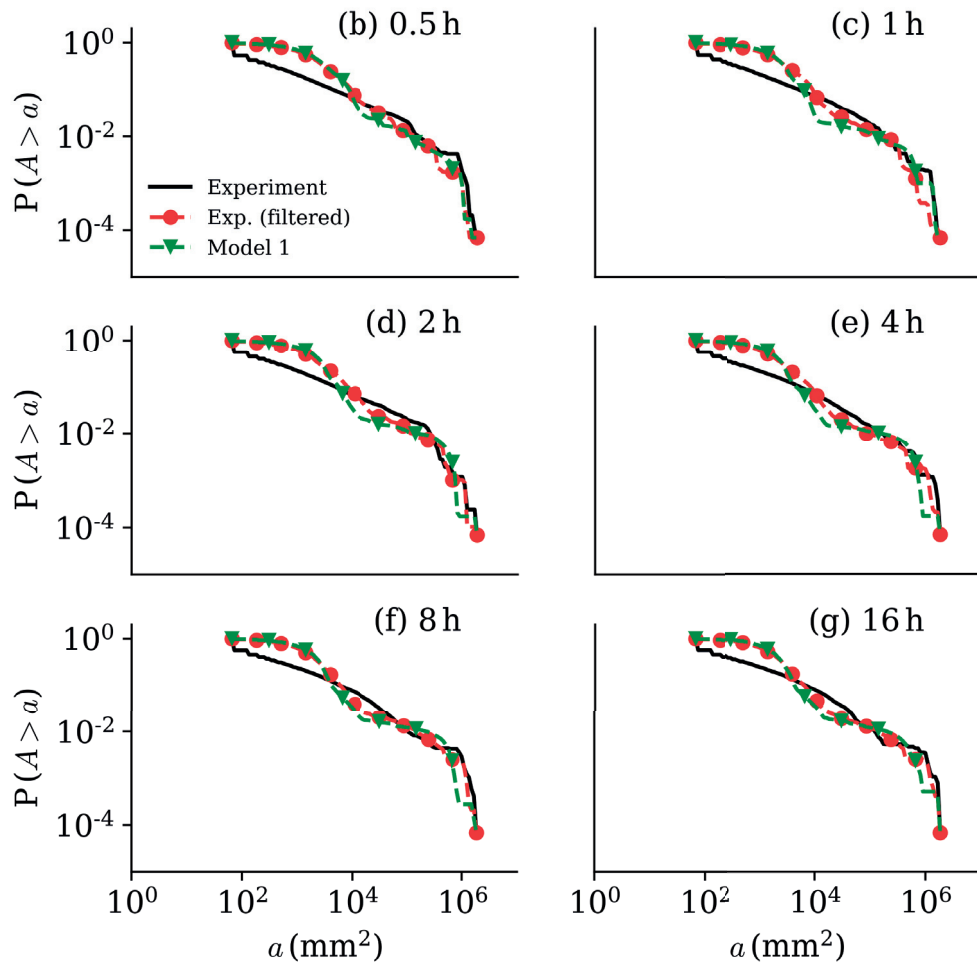


Figure 4.10 – Exceedance probability of drainage area for the model and experiment. The model is able to reproduce the filtered experimental data.

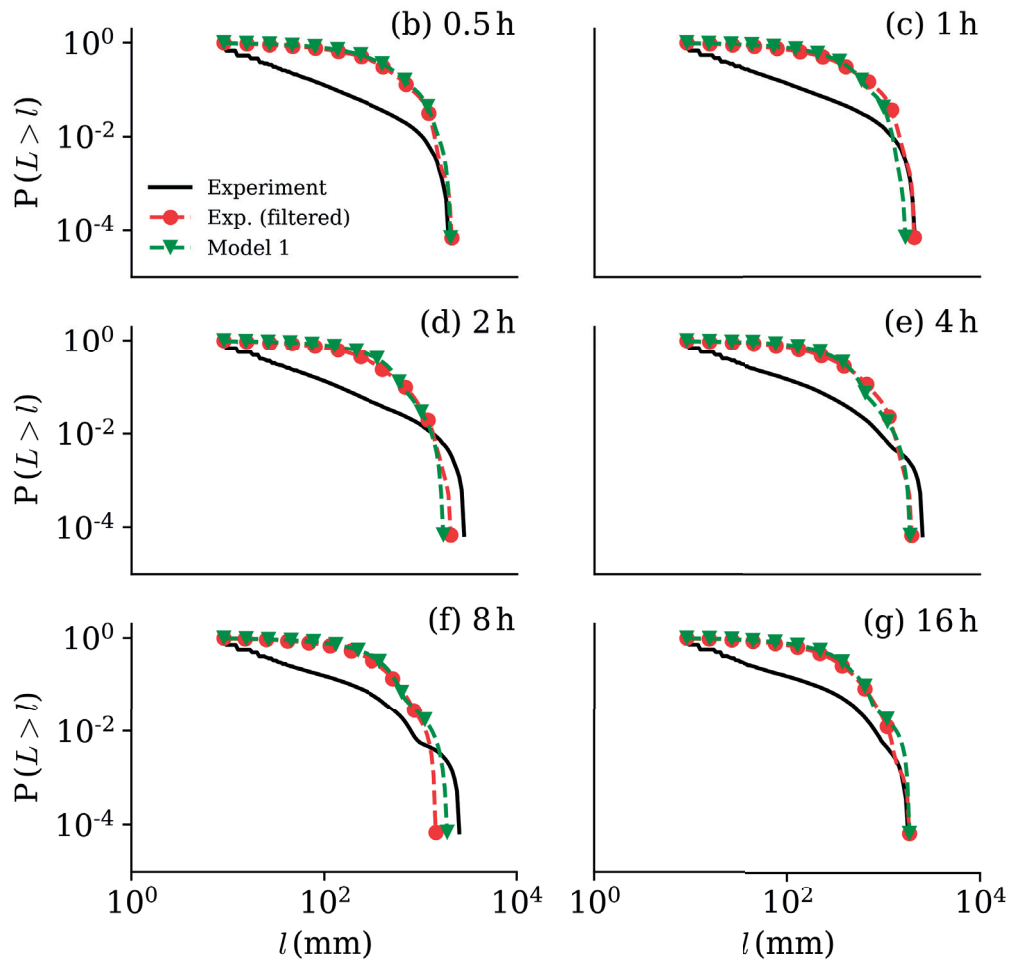


Figure 4.11 – Exceedance probability of upstream length for the model and experiment. The model is able to reproduce the filtered experimental data.

4.5 LEM modification to enhance the statistical features

The original LEM (equation 1.16) is named Model 1 and is extensively described in section 4.4. In this section, different approaches of model modification are described which are labeled as Models 2-8. The results of the models are compared in section 4.6.

4.5.1 Spatially non-uniform diffusion based on precipitation distribution (Model 2)

The first hypothesis is that the diffusion effect is a function of the droplet impacts on the surface. In other words, a higher precipitation rate causes greater smoothness on the surface. Therefore, the original LEM (equation 1.16) converts to:

$$\begin{cases} \frac{\partial z}{\partial t} = \nabla(D\nabla z) - K f(Q^m S - \Omega_{cr}) \\ D(x, y) = \bar{D} R^*(x, y) \end{cases} \quad (4.2)$$

where $D(x, y)$, $R^*(x, y)$, K and Ω_{cr} are the diffusion coefficient, normalized (based on the mean) rainfall distribution (Figure 3.4-h), convective coefficient, and critical stream power, respectively. The terms and parameters are defined in more detail in section 1.3.3. The parameters \bar{D} , K and Ω_{cr} are calibrated to see the effect of a non-uniform distribution of the diffusion coefficient on the model performance.

4.5.2 LEM as a stochastic partial differential equation (Models 3-7)

4.5.2.1 Roughness probability distribution function

The analyses in section 4.4.2 show that the statistical differences between the numerical and experimental results stem from the surface roughness. To select an appropriate probability distribution function for the random variables, we calculate the roughness distribution of the experimental results. The roughness field is calculated as the difference between the original and filtered morphology (using the Blackman window, section 4.4.2) and plotted in Figure 4.12. As expected, the majority of roughness values are in the range of sediment particle sizes. The outliers (values

more than 95 %) are removed from the data. Afterward, the data are normalized with the maximum value (95 % percentile = 10.57 mm). The probability distribution of the normalized roughness for all scans is shown in Figure 4.13. In both Figures 4.12 and 4.13, the PDFs are normalized such that $\sum_{i=1}^{100} PDF(i)\delta x_i = 1$, where δx_i is the i th bin size. The data of Figure 4.13 were fitted with 77 different standard probability distribution functions. The calibration was based on the root-mean square error between the standard PDFs and experimental data. As shown in Figure 4.14, the best fit is the “Half-Cauchy” distribution [*Cauchy*, 1853] in the form of:

$$PDF(\eta) = \frac{2\gamma}{\pi(\eta^2 + \gamma^2)} \quad (4.3)$$

where $\gamma = 0.1175$. The fitted distribution is shown in Figure 4.13. In the following analysis, this distribution is used to include the spatially and temporally random parameters in the LEM.

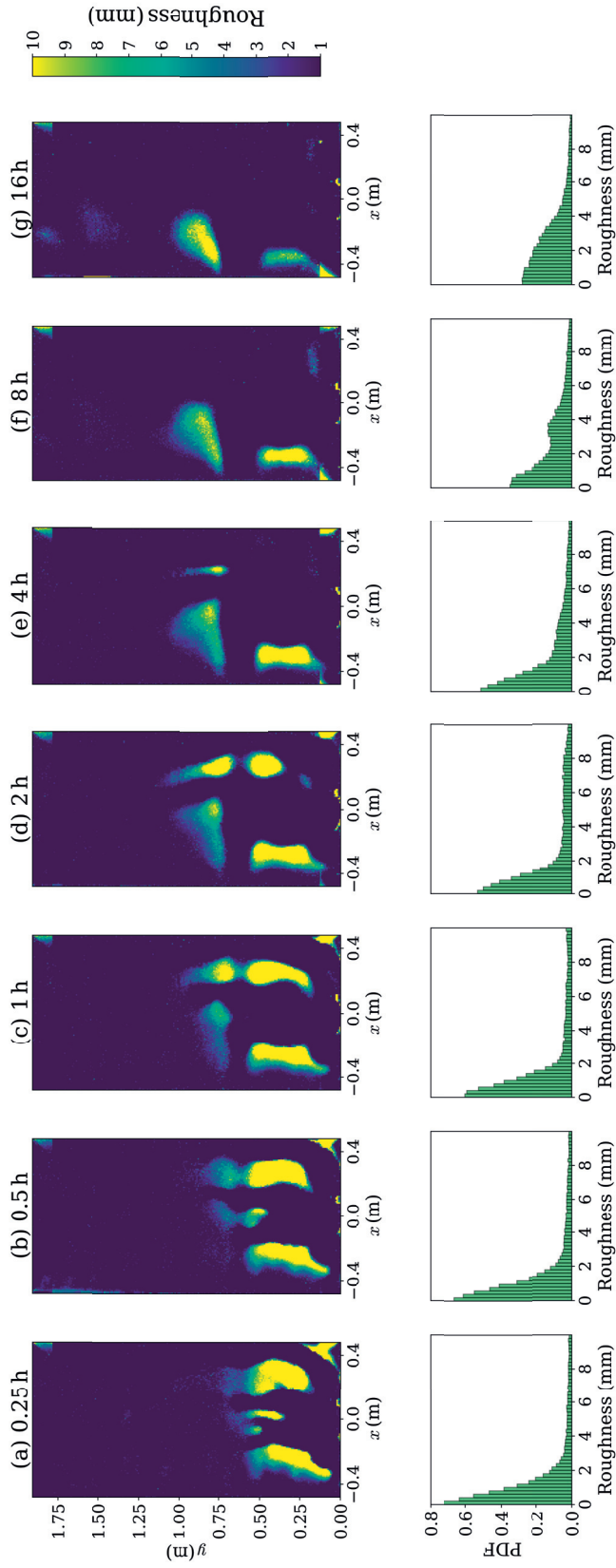


Figure 4.12 – The roughness distribution at different times ($t = 0.25, 0.5, \dots, 16h$). Roughness is defined as the difference between morphology and its filtered value using the Blackman window (section 4.4.2).

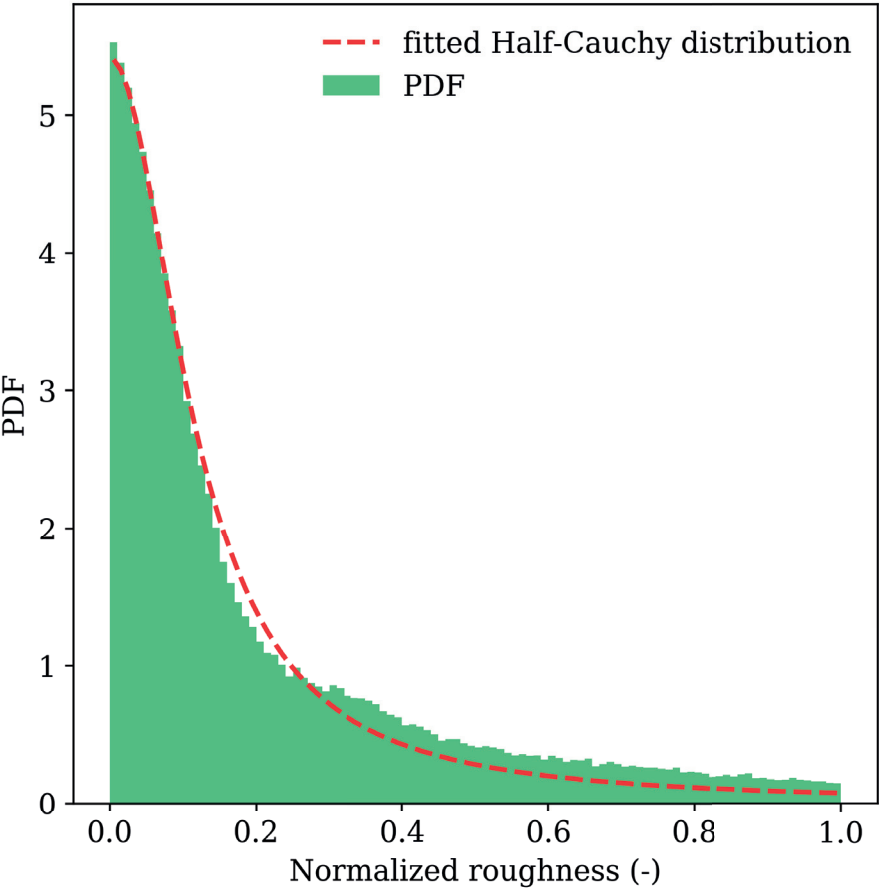
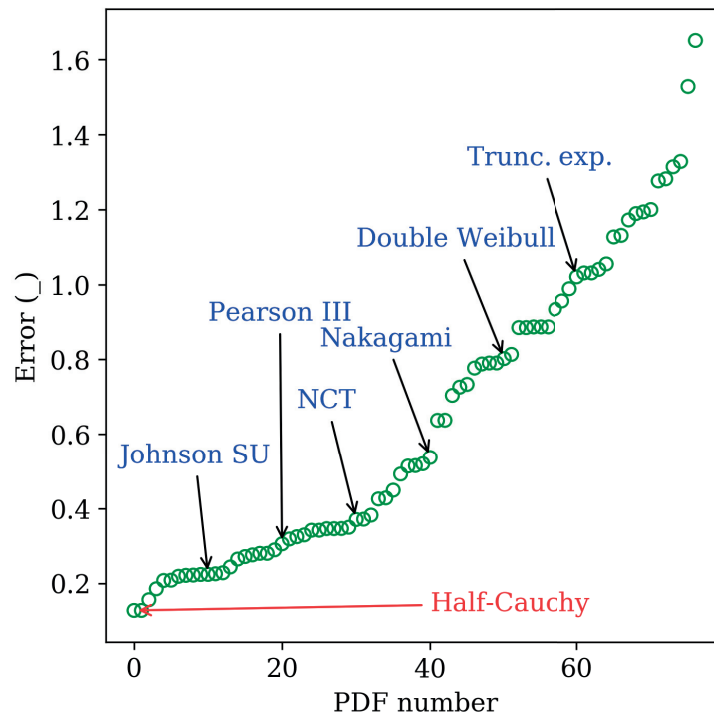


Figure 4.13 – The probability distribution of roughness during the 16-h course of the experiment. The outliers (more than 95% percentile) are removed from the dataset, and the values are normalized based on the maximum roughness.

4.5. LEM modification to enhance the statistical features



1. Half-Cauchy	2. Fold-Cauchy	3. Power log-normal	4. Reciprocal inv. Gaussian
5. Fisk	6. Fatigue-life	7. Beta prime	8. Johnson SB
9. Pareto	10. Log-normal	11. Johnson SU	12. Lomax
13. Generalized Pareto	14. Generalized gamma	15. Inverse Gaussian	16. Chi2
17. Gamma	18. Frechet-r	19. Weibull-min	20. Exponentiated Weibull
21. Pearson III	22. Alpha	23. Log-laplace	24. Burr
25. Generalized exponential	26. Exponential	27. Inverse Weibull	28. Generalized extreme
29. Inverse gamma	30. Gilbrat	31. NCT	32. Gauss hypergeometric
33. Wald	34. Levy	35. Exponential power	36. Bradford
37. NCF	38. Chi	39. NCX2	40. Beta
41. Nakagami	42. Generalized half-logistic	43. Half-logistic	44. F
45. Gompertz	46. Cauchy	47. Tukey-lambda	48. T
49. Fold-normal	50. Half-normal	51. Double Weibull	52. Double gamma
53. Laplace	54. generalized logistic	55. Gumbel-r	56. Frechet-l
57. Weibull-max	58. Kolmogorov-Smirnov	59. Triangular	60. Hyperbolic secant
61. Truncated exponential	62. Rayleigh	63. Rice	64. Logistic
65. Maxwell	66. Normal	67. Log-gamma	68. Arcsine
69. Truncated normal	70. Rdist	71. Power law	72. Gumbel-l
73. Anglit	74. Uniform	75. Semicircular	76. Levy-l
77. Cosine			

Figure 4.14 – The probability distribution functions that were calibrated with the roughness distribution in Figure 4.13. The PDF numbers (1-77) are assigned based on their fitness with the experimental roughness (error value in the plot). The error value is the root mean square error between the experimental distribution and calibrated PDF. The distribution names are taken from the Python library.

4.5.2.2 Model assumptions

Here we assume a heterogeneous diffusion coefficient (D) and add a roughness term ($M\eta$) to the original model (equation 1.16). The modified LEM is in the form of:

$$\frac{\partial z}{\partial t} = \nabla \cdot (D \nabla z) - K f(Q^m S - \Omega_{cr}) \pm M\eta \quad (4.4)$$

where D , K and Ω_{cr} are the diffusion coefficient, advection coefficient and critical stream power, respectively. In the roughness term ($M\eta$), η is a random variable from the Half-Cauchy probability distribution function of roughness (section 4.5.2.1, equation 4.3) and M is calibrated based on the experimental data. In Table 4.2, different simulation scenarios are presented depending on the conditions of D , K , Ω_{cr} and η . In Models 3, 4, 5, random values of D , K , Ω_{cr} are assumed, respectively while there is no roughness term in the model ($M = 0$). In Model 6, parameters D , K and Ω_{cr} are spatially uniform, and a random roughness term ($M\eta$) is considered. Finally, a spatially random diffusion coefficient (D) is coupled with the roughness term in Model 7 such that higher roughness is associated with lower values of D .

Table 4.2 – Model assumptions for different stochastic PDEs. The parameter η is a random variable from the Half-Cauchy probability distribution function (equation 4.3).

	D (mm ² h ⁻¹)	K (mm ^{-1/2} h ^{-1/2})	Ω_{cr} (mm ^{3/2} h ^{-1/2})	roughness term (mm h ⁻¹)
Model 3	$D_{max}(1 - \eta)$	unifom	unifom	0
Model 4	unifom	$K_{max}(1 - \eta)$	unifom	0
Model 5	unifom	unifom	$\Omega_{cr}^{max}\eta$	0
Model 6	unifom	unifom	unifom	$M\eta$
Model 7	$D = D_{max}(1 - \eta)$	unifom	unifom	$M\eta$

4.5.3 Diffusion control via a critical curvature (Model 8)

As described in section 4.2, the PDE was solved in two steps. The first step was to solve the convective part of the LEM via Runge-Kutta method. Then, the diffusion part was solved. The problem with a uniform diffusion coefficient was that it applied the same rate of smoothing on the whole domain. In this section, a deterministic approach is introduced to selectively apply the diffusion (smoothing) part of the equation. Using

4.5. LEM modification to enhance the statistical features

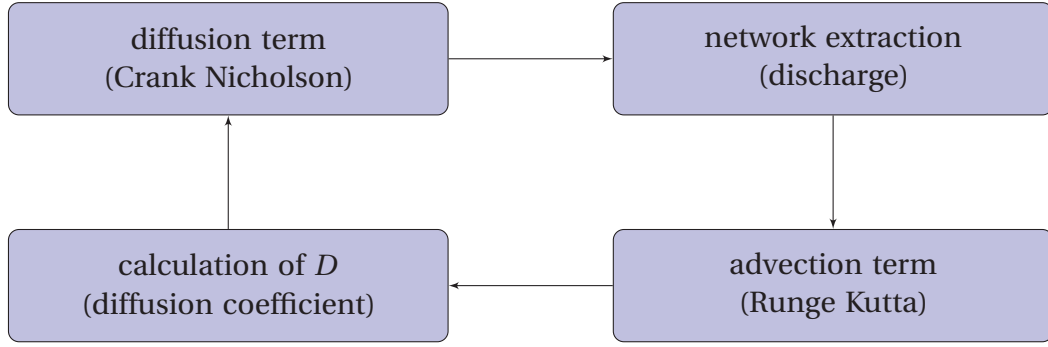


Figure 4.15 – Calculation of the diffusion coefficient at each time step, after solving the advection term and before applying the diffusion.

the fact that the incised parts of morphology have a larger curvature, we defined the diffusion coefficient as a function of local curvatures. The values for D are calculated after solving the advective part of the LEM (Figure 4.15). By this approach, the model will have more smoothing effects on the incised parts of the surface. The model is defined in the form of:

$$\frac{\partial z}{\partial t} = \nabla \cdot (D \nabla z) - K f(Q^m S - \Omega_{cr}) \quad (4.5)$$

with

$$D(x, y, t) = \phi H(|\nabla^2 z^*(x, y, t)| - C_{cr})$$

where $H(\zeta)$ is the Heaviside step function, and $z^*(x, y, t)$ is the morphology after solving the advective term. The new parameters, C_{cr} and ϕ are the critical curvature and diffusion coefficient at high curvatures, respectively, which are calibrated via the experimental data.

4.5.4 Parameter optimization of the modified models

A multi-objective evolutionary algorithm (Borg MOEA) [Hadka and Reed, 2013] was used to find the optimum parameters. The initial condition of the numerical modeling was the experimental data at $t = 0.25$ h (Figure3.4a). Models 2-8 were calibrated using two objective functions, f_1 and f_2 , which are based on the morphology (z) and the

exceedance probability of upstream length ($P(L > l)$), respectively:

$$f_1 = \sqrt{\frac{\sum_{i=1}^N (z_{i,model} - z_{i,exp})^2}{N}}|_{t=8h} \quad (4.6)$$

$$f_2 = 10^{(RMSE(0.25) + RMSE(0.5) + \dots + RMSE(8))}$$

with

$$RMSE(x) = \sqrt{\frac{\sum_{j=1}^{100} (P_{model}(L > l_j) - P_{exp}(L > l_j))^2}{100}}|_{t=x} \quad (4.7)$$

where $z_{i,model}$ and $z_{i,exp}$ are model and experimental elevation at cell i ; Moreover, $P_{model}(L > l_j)$ and $P_{exp}(L > l_j)$ represent the exceedance probability of upstream length for model and experiment at l_j (l is divided into 100 equally spaced points at the logarithmic scale). Note that only the power law parts of the exceedance probabilities were compared (linear part of Figure 3.10c where $P(L > l) = l^{-\xi}$).

In brief, f_1 defines the root mean square error of the morphology (z) at $t = 8$ h and f_2 represents the summation over the root mean square errors of the upstream length distribution at $t = 0.25, 0.5, \dots, 8$ h. The analysis showed that the model results for the three scaling laws (Hack's law and the exceedance probabilities of drainage discharge and upstream length) were correlated, i.e., once the model yields the optimized value for f_2 , the error for the rest of scaling laws were minimized.

4.6 Results of the modified models

In this section, Models 2 through 8 are compared to find out the best modification approach(s) for the original LEM (Model 1). The calibrated parameters of Models 2 (the diffusion coefficient correlated with the rainfall distribution, equation 4.2), Models 3-7 (the stochastic PDEs, equation 4.4) and Model 8 (the diffusion coefficient as a function of curvature, equation 4.5) are presented in Tables 4.3, 4.4 and 4.5, respectively.

In order to compare the performance of the calibrated models, the simulated morphologies at $t = 8$ h are shown in Figure 4.16. All the models are correctly showing the main direction of morphology evolution (toward the right side of the domain where the precipitation rate is higher). However, solely based on the elevation field,

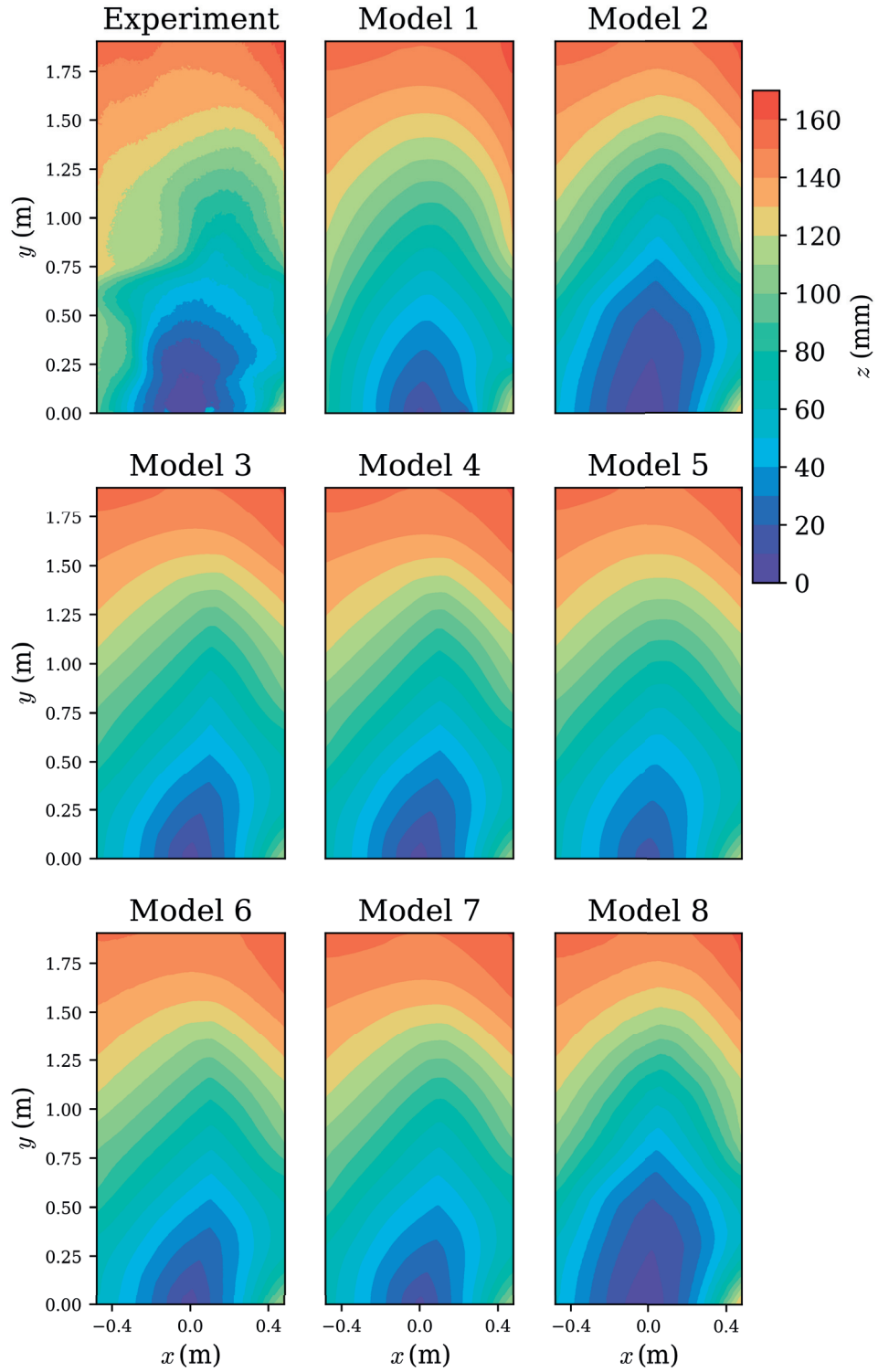


Figure 4.16 – Elevation field (z) of the models at $t = 8$ h. Model 1 is the original LEM (equation 1.16). In Model 2, the diffusion coefficient is correlated with the rainfall distribution. Model 3 to 7 are the stochastic PDEs which are defined in Table 4.2. In Model 8, the diffusion coefficient is a function of curvature (equation 4.5).

Chapter 4. Landscape evolution model in absence of rills

Table 4.3 – Calibrated parameters for Model 2. The diffusion coefficient at each cell ($D(x, y)$) is proportional to the rainfall intensity, $R^*(x, y)$. The spatially averaged diffusion coefficient is shown by \bar{D} .

	\bar{D} ($\text{mm}^2 \text{h}^{-1}$)	K ($\text{mm}^{-\frac{1}{2}} \text{h}^{-\frac{1}{2}}$)	Ω_{cr} ($\text{mm}^{\frac{3}{2}} \text{h}^{-\frac{1}{2}}$)
Model 2 ($D = \bar{D}R^*(x, y)$)	7360	0.258	51.19

Table 4.4 – Calibrated parameters for the stochastic PDEs (Models 3-7). The spatially random coefficients are defined according to Table 4.2.

	D_{max} ($\text{mm}^2 \text{h}^{-1}$)	K_{max} ($\text{mm}^{-\frac{1}{2}} \text{h}^{-\frac{1}{2}}$)	Ω_{max} ($\text{mm}^{\frac{3}{2}} \text{h}^{-\frac{1}{2}}$)	M (mm h^{-1})
Model 3 (random D)	20518	0.414	49.09	–
Model 4 (random K)	14036	0.628	41.92	–
Model 5 (random Ω_{cr})	17052	0.200	34.20	–
Model 6 (random η)	15921	0.387	49.85	268.6
Model 7 (random D and η)	19131	0.270	33.71	44.8

it is not possible to have an accurate model comparison. For detailed scrutiny of the surfaces, the corresponding power spectral densities at $t = 8 \text{ h}$ are shown in Figure 4.17. As can be seen, the high PSD values of the models and experiment are in the same range whereas more differences are detected at high wave numbers. The azimuthally averaged PSD in Figure 4.18 reveals that Models 2, 3, 4 and 5 do not have the high wavenumber structure of the experimental data. Indeed, no improvements are detected in these models as the PSD values are worse than the values in the original model (Model 1). Model 6 (the PDE with a random roughness term) also has a lower PSD value compared to Model 1, except for the highest wave numbers. On the other hand, Model 8 shows a better agreement with the measured morphology.

The extracted discharge networks from the morphologies are shown in Figure 4.19. According to the discharge fields, the low-order branches of the experimental network are not captured by models 2, 3, 4, 5 and 7. As shown in the spectral analysis, these models smooth the initial morphology and remove the low-discharge points of the experiment. Models 6 (with a random roughness term) and 8 (with diffusion as a step function of curvature) generate more accurate discharge networks at the low-discharge areas. This comparison shows that applying the spatiotemporal random values for the variables D , K and Ω_{cr} does not improve the original deterministic model (Model 1). In other words, if we assume the model as a stochastic PDE, adding a stochastic roughness term ($M\eta$) is crucial. In the following, we will go through models 6, and 8 which are the best scenarios.

4.6. Results of the modified models

Table 4.5 – Optimized parameters for Model 8. The diffusion coefficient is defined as a step function of curvature (equation 4.5, Figure 4.15).

	ϕ (mm ² h ⁻¹)	C_{cr} (mm ⁻¹)	K (mm ^{-$\frac{1}{2}$} h ^{-$\frac{1}{2}$})	Ω_{cr} (mm ^{$\frac{3}{2}$} h ^{-$\frac{1}{2}$})
Model 8 (conditional D)	17728	0.002916	0.180	10.32

Morphology evolutions via Models 6 and 8 are compared with the experimental data and the original LEM (Model 1) in Figure 4.20. Both Models 6 and 8 can simulate the main direction of the evolution from left to right. Compared to Model 1, Model 6 shows less evolution of the low elevation area ($z < 70$ mm) to the upstream whereas Model 8 results in more upward extension of the low elevation area.

In Figure 4.21, the discharge fields of Models 6 and 8 are compared with both the experimental data and the original LEM. As previously mentioned, both Models 6 and 8 modify the discharge network of the original LEM by providing a more branched structure at the low-discharge region. At the initial stages of erosion ($t = 0.5$ h and 1 h), the network of Model 2, is more similar to the experimental data, as compared to Model 8. Nonetheless, Model 6 generates a straight mainstream at $t = 2$ h which is different from the experimental data. Furthermore, the upstream branches of the network at $t = 4$ h disappear at $t = 8$ h, and reappear at $t = 16$ h. On the other hand, Model 8 renders a network which converges slower and shows a more consistent level of details at the low-discharge areas.

In order to further confirm the qualitative descriptions, the relation between the upstream length (l) and drainage area (A) is presented in Figure 4.22. Based on the analysis of Chapter 3, the experimental data make a temporally-invariant power-law relation between l and A (Hack's law). Figure 4.22 shows that the original LEM (Model 1) has a huge deviation from the experimental data. However, the scenarios 6 and 8 improve the LEM to have a better statistical representation of the experiment. Model 8 appears to be even better than Model 6 at the upstream (low-drainage areas). The exceedance probabilities of discharge, drainage area, and upstream length are shown in Figures 4.23-4.25, respectively. According to Figures 4.23 and 4.24, Models 6 and 8 do not show a significant improvement in the exceedance probabilities of drainage discharge and area. The exceedance probability of upstream length (Figure 4.25) shows that for smaller lengths, Model 8 has a better agreement with the experimental data. However, Model 6 shows a better performance than Model 8 at higher values of l . Overall, these results demonstrate that via adding a random roughness term (Model 6) and having a step function for the diffusion coefficient

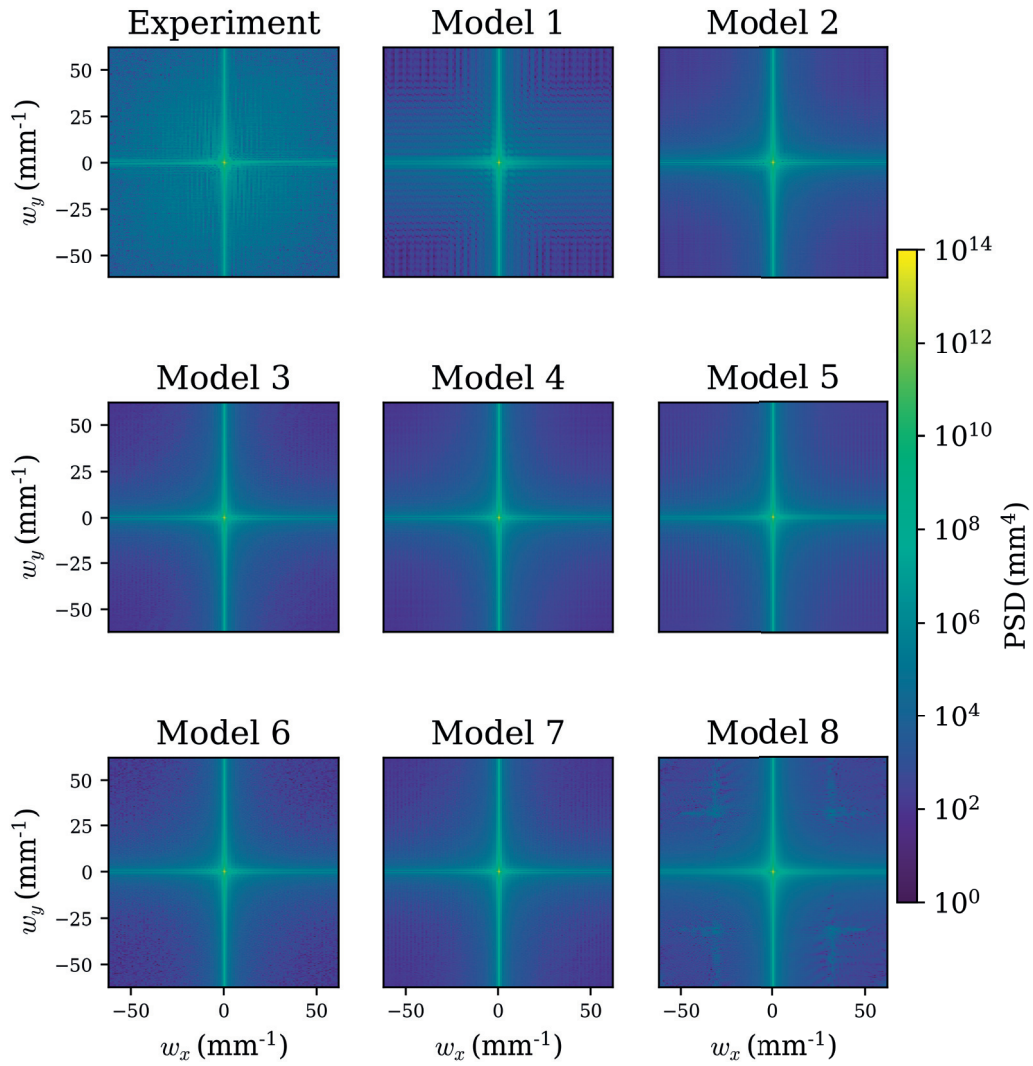


Figure 4.17 – Power spectral density of the elevation fields (z) at $t = 8\text{h}$ at the logarithmic scale. Model 1 is the original LEM and Models 2 to 8 are defined in section 4.5. The low-wave number PSD values of the models and experiment are in the same range whereas the models behave differently at high wave numbers.

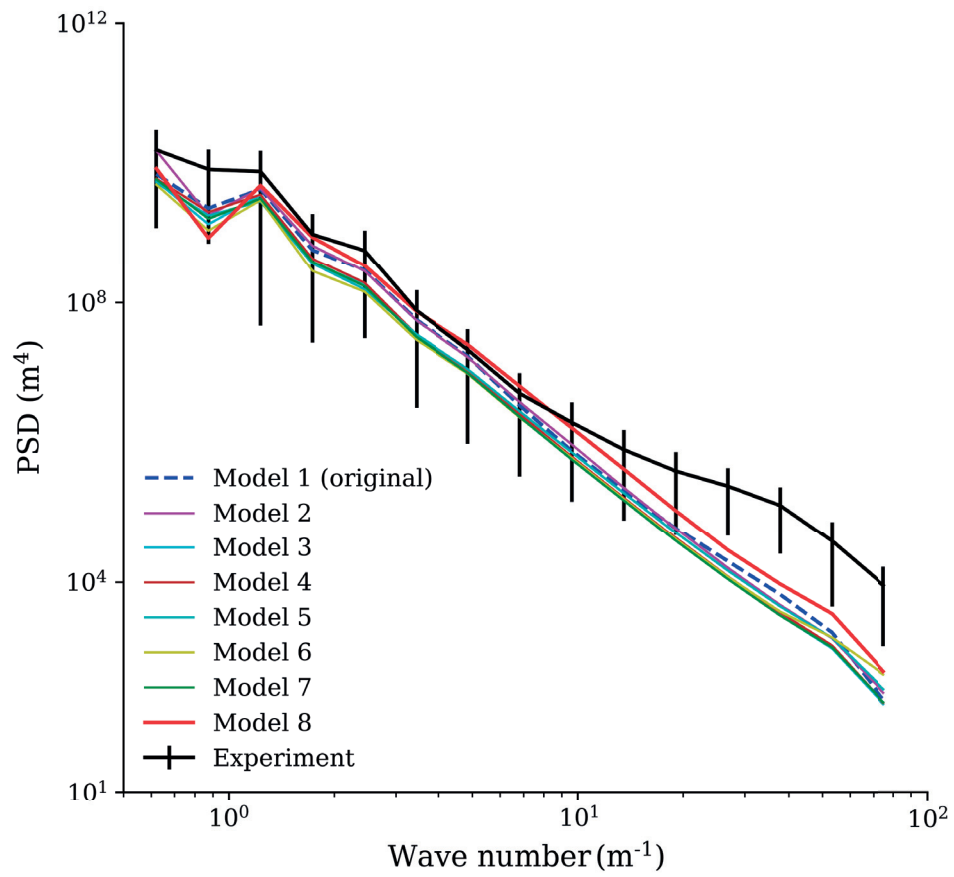


Figure 4.18 – The azimuthally-averaged power spectral density of the morphology in the wave number domain. The errorbars of the experimental data are based on the standard deviation from the mean. Model 1 is the original LEM and Models 2 to 8 are defined in section 4.5.

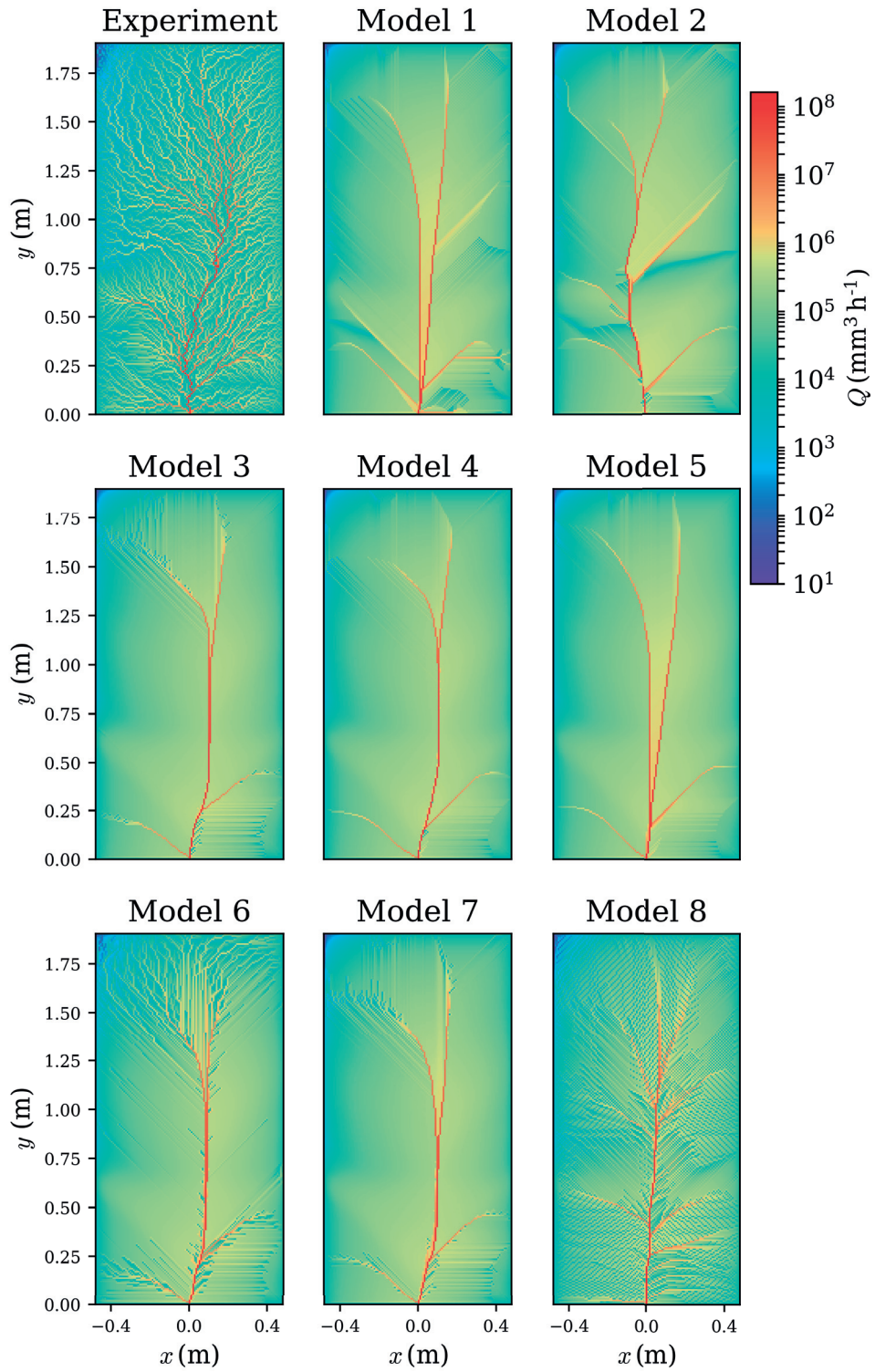


Figure 4.19 – Discharge distribution of the experiment and different models at $t = 8 \text{ h}$. Model 1 is the original LEM and Models 2 to 8 are defined in section 4.5.

(Model 8), more micro-roughnesses of the morphology can be simulated by the LEM. Nonetheless, the statistical characteristics are not completely matched with the experimental data.

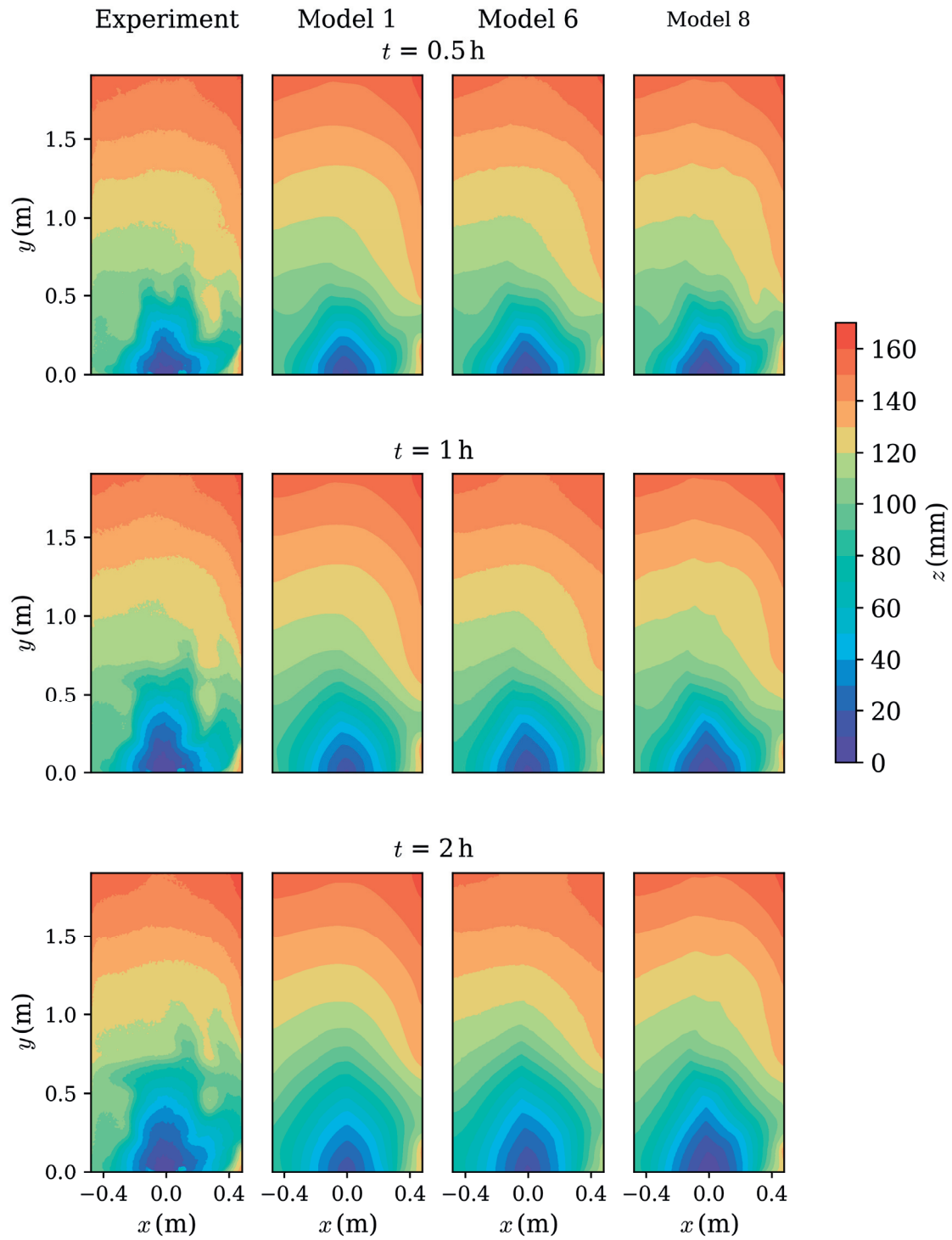


Figure 4.20 – Continued

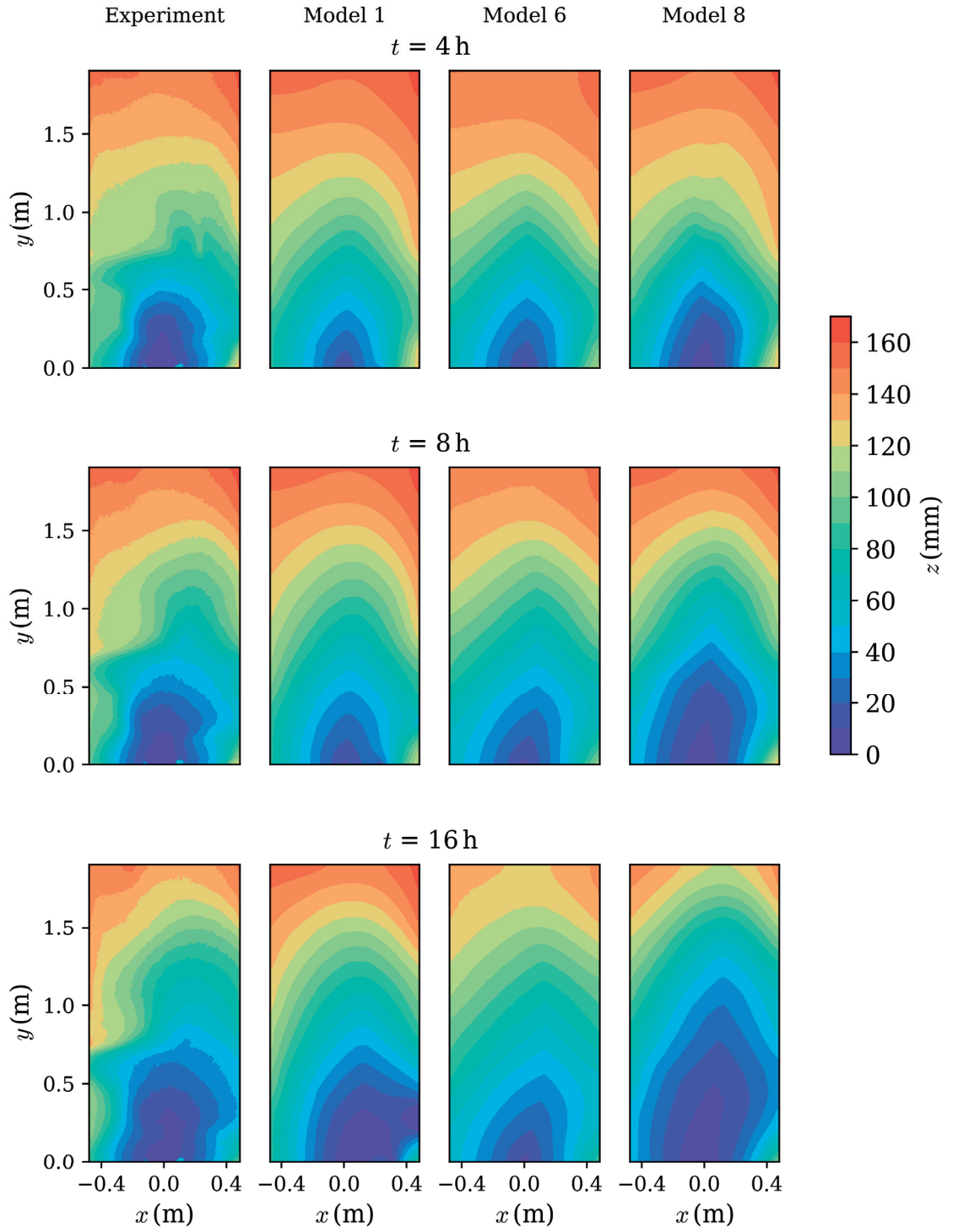


Figure 4.20 – Elevation field of the experiment and Models 1, 2 and 8. Model 1 is the original LEM and Models 2 and 8 are the modifications defined in section 4.5.

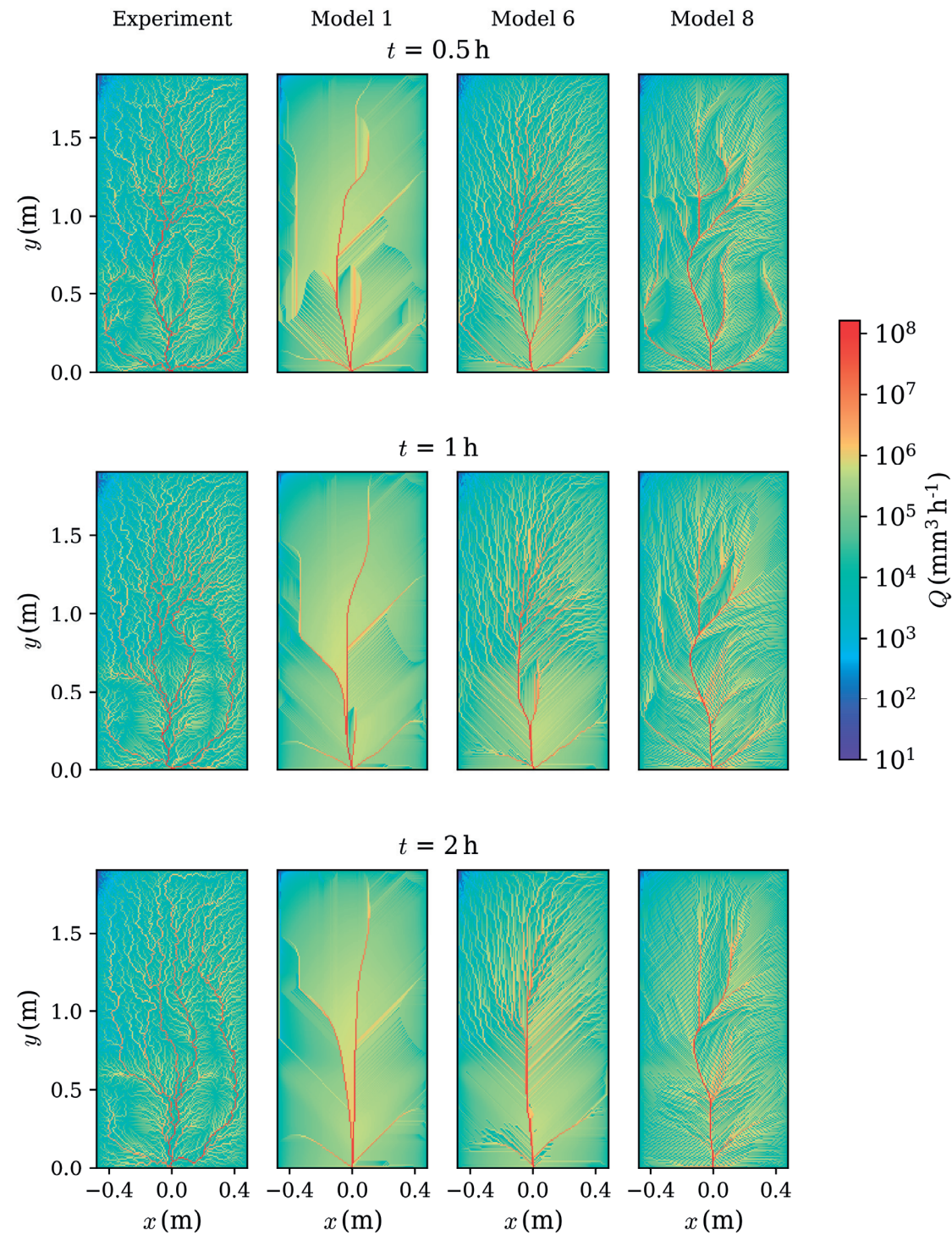


Figure 4.21 – Continued.

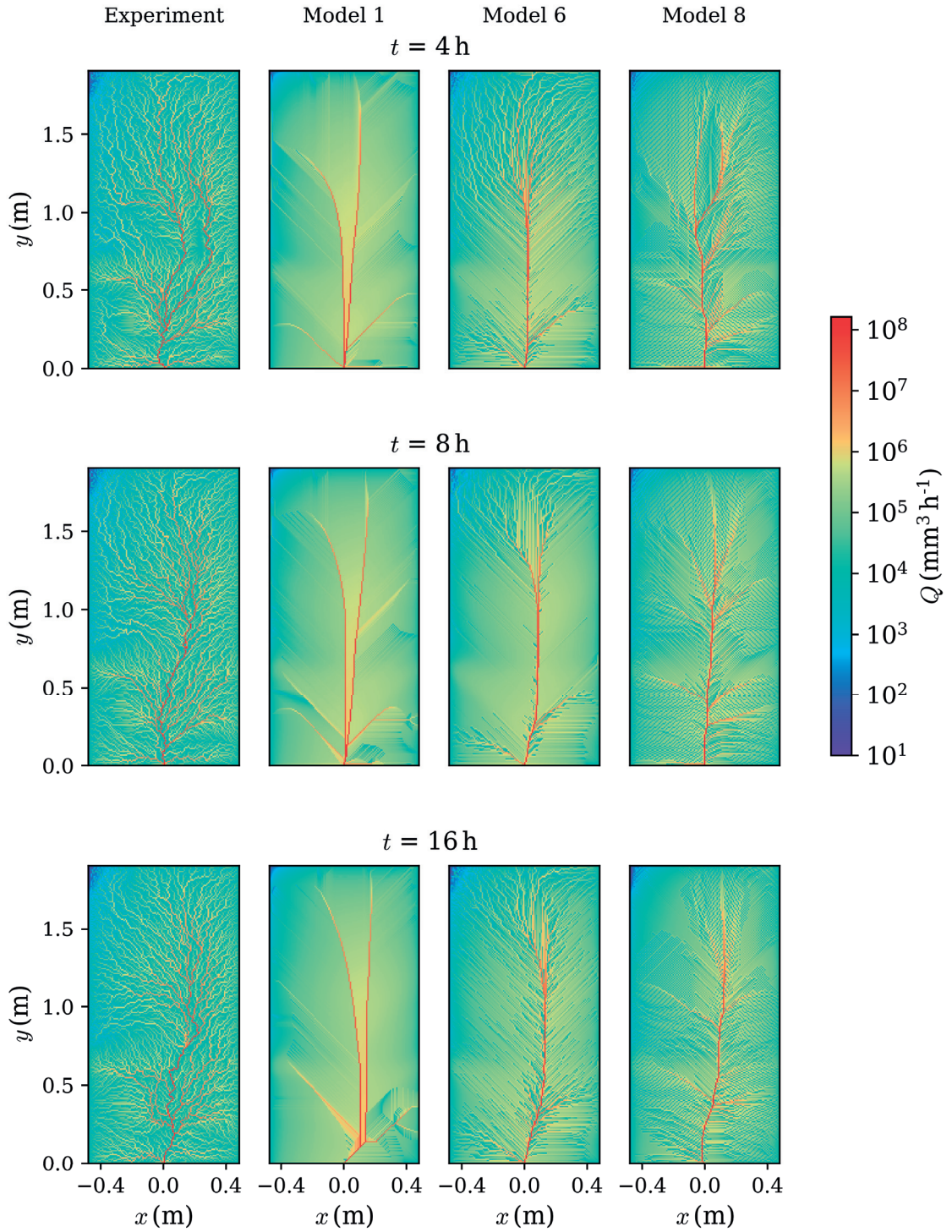


Figure 4.21 – Discharge distribution of the experiment and Models during the 16-h course of experiment. Compared to the original LEM, Models 6 and 8 are more in agreement with the experiment by producing a more branched network at the low-discharge.

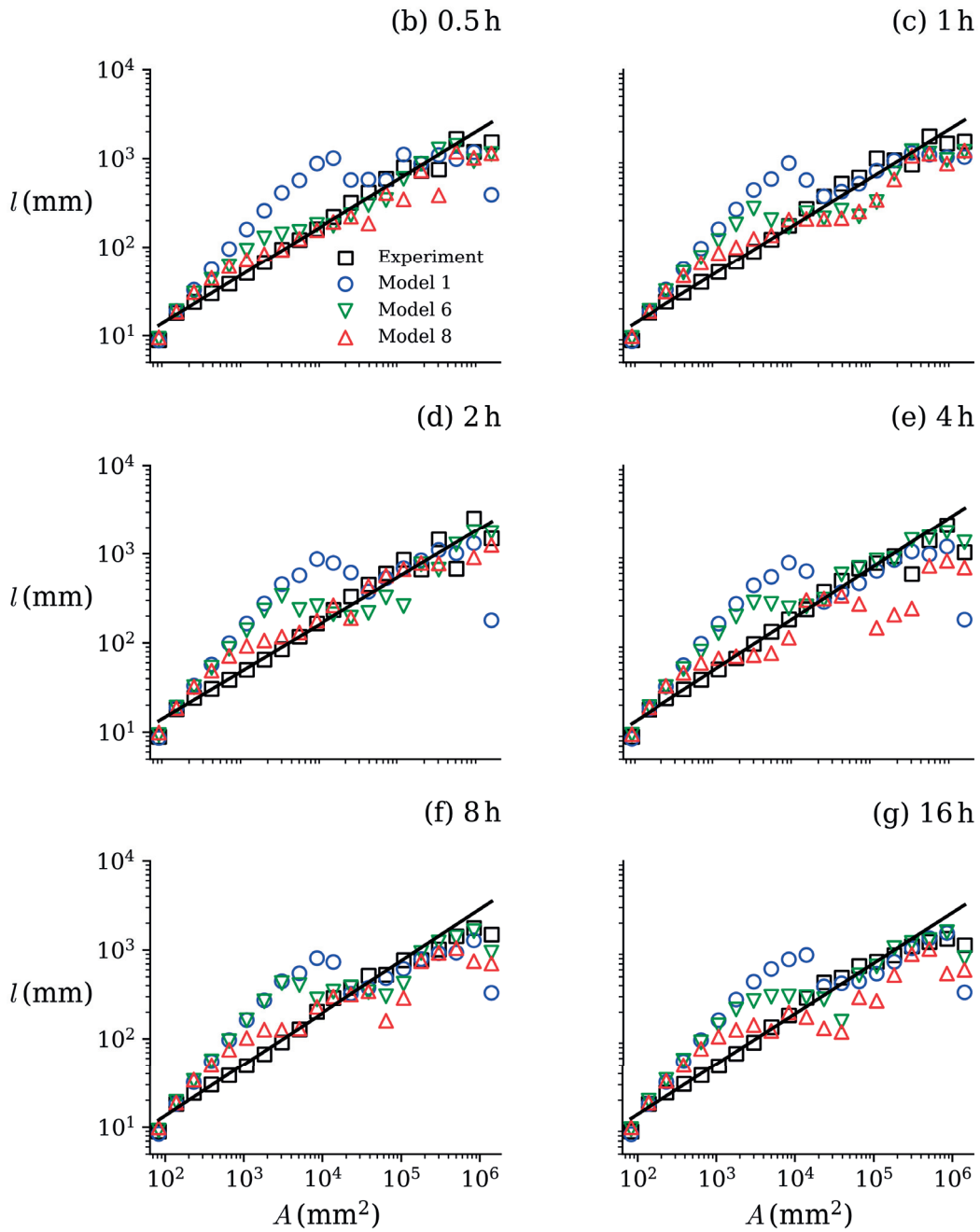


Figure 4.22 – The relation between the upstream length and drainage area (Hack's law) for the models and experiment. Models 6 and 8 improve the original LEM to follow Hack's law. However, Model 8 shows more enhancement compared to Model 6.

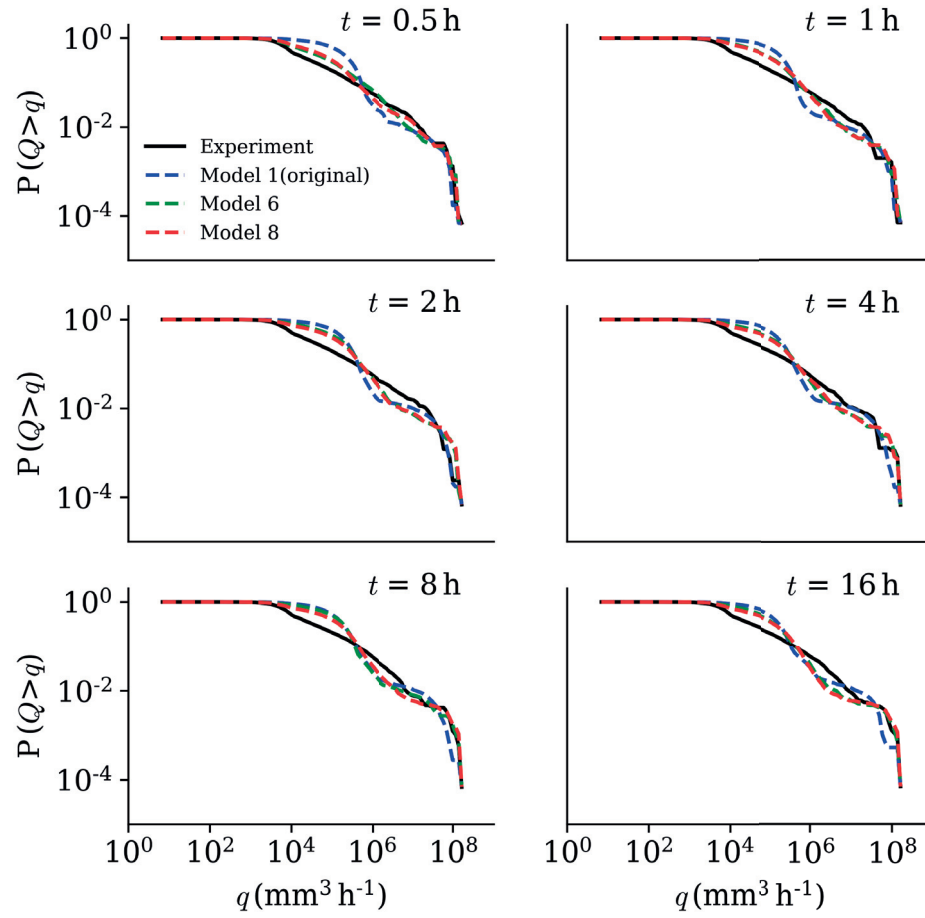


Figure 4.23 – Exceedance probability of drainage discharge for the experiment and Models 1, 6 and 8.

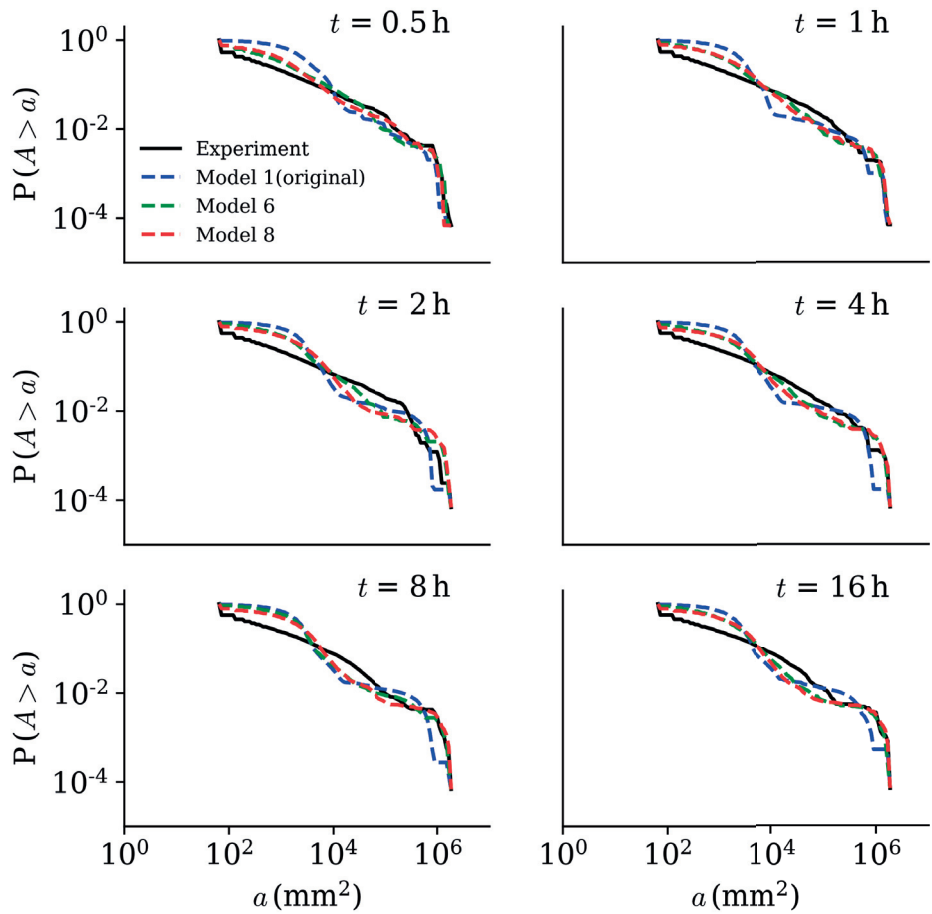


Figure 4.24 – Exceedance probability of drainage area for the experiment and Models 1, 6 and 8.

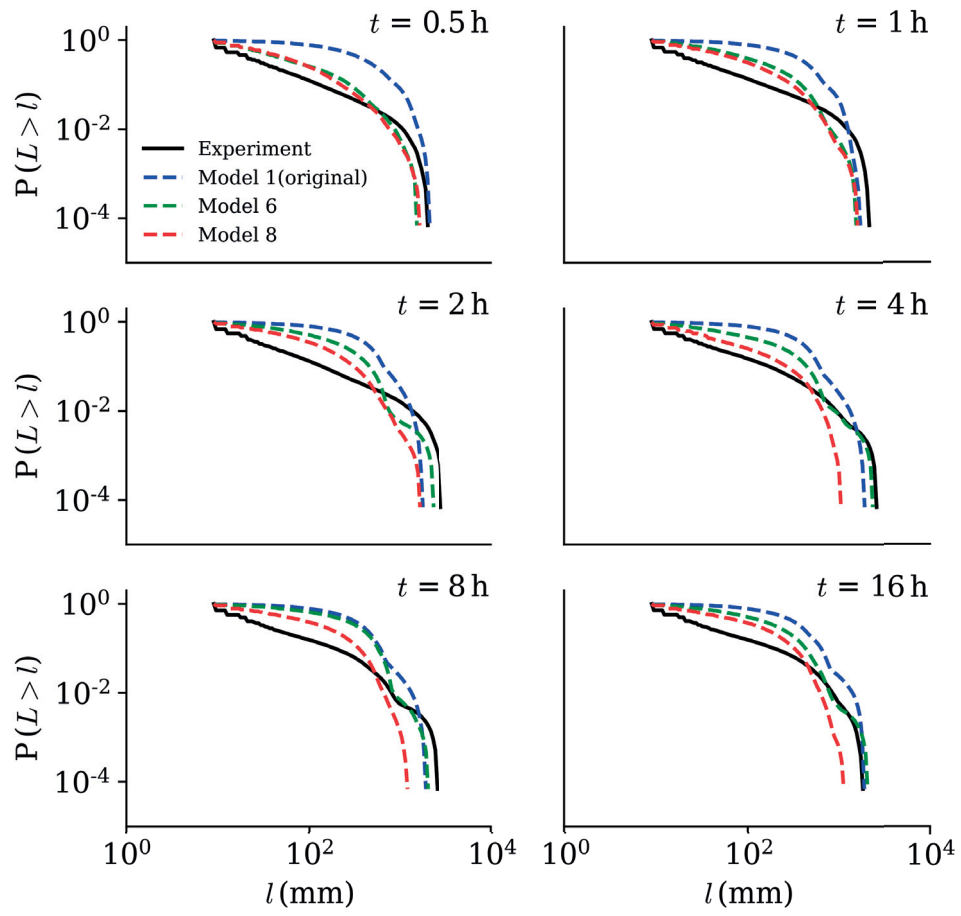


Figure 4.25 – Exceedance probability of upstream length for the models and experiment.

4.7 Conclusions

This study showed that the large scale LEM was a robust simulation tool in the absence of rills (previous investigations focused on incised landscapes), when coupled with the D8 algorithm for shallow, overland flow. The LEM was able to capture the morphology changes, which evolved under a highly heterogeneous rainfall, by calibrating only three parameters, i.e., diffusion coefficient (D), advection coefficient (K) and the critical stream power (Ω_{cr}). Erosive processes captured by the LEM are broadly characterized as raindrop impact-induced diffusion and surface flow-induced shear stress. Unlike previous investigations, here the surface was continuously covered by overland flow, which was modeled as a flow network. Shear stress-driven erosion did not occur until a soil-specific flow rate was exceeded. It was also revealed that even at a diffusive-dominated area where the morphology was not incised, the advective term was inevitable to define the right evolution direction. The intricate surface flow details produced by the measured surface were not present in the modeled surface. Filtering of the measured and modeled surfaces markedly improved the agreement between the calculated flow networks. However, the scaling laws reported in Chapter 3 were not maintained. In general, the LEM with a uniform diffusion coefficient could reproduce the low-pass filtered experimental results. In other words, small scale variability due to surface roughness that increased over the course of the experiment was not present in the LEM, and thus had to be removed to obtain reasonable statistical agreement between the experimental measurements and the LEM.

Different methods were tested to improve the applicability of the LEM further. The objective was to introduce some methods that preserve the simplicity of the standard advective-diffusive LEM. In the first approach, the diffusion coefficient was correlated with the rainfall distribution (Model 2). Afterward, the LEM was solved as a stochastic PDE (Models 3-7). Finally, the diffusion coefficient was calculated as a step function of curvature (Model 8). The analyses clarified that assuming a spatially non-uniform diffusion coefficient correlated with the precipitation rate did not improve the model results. Similarly, using a random distribution of the diffusion coefficient (D), convection coefficient (K) and critical stream power (Ω_{cr}) could not enhance the model quality. In these models, the high value (or gradient) of the diffusion coefficient smoothed the morphology, regardless of the uniformity or heterogeneity of the other parameters. Based on numerical simulations calibrated

with high-resolution experimental results, this study proposed two straightforward ways of LEM improvement: I) adding a spatiotemporal random roughness term in the LEM and II) applying the diffusion term as a function of curvature. By using these two approaches, the relation between the upstream length and drainage area of the model(Hack's law) was improved, and more details of the low-discharge region of the experimental networks emerged in the model results. However, insignificant enhancements were observed in the exceedance probabilities of drainage area and upstream length.

5 Conclusions and outlook

5.1 Summary and conclusions

Morphological features are undeniable in environmental transport processes as they provide the geometry of the systems. Today's advanced environmental analyses with high-resolution data require more accurate criteria for surface evolution modeling. This dissertation involved experiment and numerical simulations of rainfall-driven sediment transport at laboratory scales where we focused on the statistical and physical characteristics of unchanneled overland flow morphologies. In the following, the primary outcomes are presented in accordance with the research questions mentioned in section 1.5.

5.1.1 Rainfall-driven hysteretic sediment fluxes as a result of morphological changes

Hysteresis loops are well-known for sediment concentration-discharge relation in river flows [*Klein*, 1984; *Williams*, 1989; *De Girolamo et al.*, 2015; *Sun et al.*, 2016; *Dean et al.*, 2016; *Sherriff et al.*, 2016]. In Chapter 2, the flume experiment showed the hysteretic erosion patterns for rainfall-driven sediment transport. Clockwise hysteresis loops occurred when a temporally variable precipitation rate was applied on the surface. Furthermore, the hysteresis loops were wider for the finer particles, and as the particle size increased, the erosion rate was more correlated with the precipitation rate (narrower hysteresis loops). The HR model [*Hairsine and Rose*, 1991, 1992a, b] was calibrated with the experimental data. The model could adequately

reproduce the concentration of finer particles. However, the results for larger particle size classes were less satisfactory. Because, large particles are more likely transported due to rolling, saltation and ejection which are not included in the HR model. The calibrated model described the generation of a shield layer due to deposition of the larger particles during the erosion phenomenon. The model results indicated that the subtle morphological changes caused a less availability for the finer sediments and thus the generation of the hysteresis loops.

5.1.2 Statistical characteristics of unchanneled morphology evolution

The hysteretic sediment fluxes in Chapter 2 demonstrated the importance of slight near-surface particle rearrangement on erosion trends. This phenomenon raised another question about the surface morphology: How are the deposited particles arranged on the unchanneled surfaces? To answer this question, we inspired from the general self-similarity concepts for river networks (Chapter 3). Despite different climates, vegetation, human activities, etc., river networks follow identical scaling laws for upstream length and drainage area [*Hack*, 1957; *Gray*, 1961; *Mueller*, 1972; *Rigon et al.*, 1996, 1998]. In our experiment, the high-resolution morphology under a spatially non-uniform rainfall was measured by a laser scanner during the experiments. The unchanneled morphologies showed the catchment-scale power laws with the same range of exponents. Consequently, self-similarity of landscapes exists at the initial stages of water-sediment interaction (rainfall) and is irrespective of sediment transport types. This investigation provided the essential statistical metrics for morphological evolution and particle deposition on the surface even when there is no visible incision on the surface.

5.1.3 Catchment-scale LEM on unchanneled morphology evolution

The remarkable statistical similarity of unchanneled surfaces with catchment-scales urged us to test the applicability of a widely used catchment-scale model [*Howard et al.*, 1994; *Perron et al.*, 2008] to laboratory-scale unchanneled surfaces. In the

previous applications of the LEM at large scales, the unchanneled areas were modeled via a diffusion term. In Chapter 4, the model with both diffusive and advective terms was solved for unchanneled surface evolution. The model was calibrated by using the high-resolution morphologies captured by a laser scanner during the 16-h experiment under a spatially non-uniform rainfall. The spectral analysis showed that with high-resolution scans of an unchanneled surface, the standard catchment-scale LEM was quite applicable to have the prevailing evolution rate and direction. The highlighted point about the LEM was its simplicity as it simulated the process with only three parameters and represented the complex overland flow by a dynamic discharge network. One issue was the high value of diffusive coefficient that caused smoothing of low-discharge areas (with high wave numbers of the Fourier transform). In other words, the model produced a spatially filtered elevation field of the experiment. In the interest of improving the statistical aspect of the model results, seven different scenarios were tested. After comparison, two approaches were suggested to relatively enhance the simulations: 1) Adding a random roughness term, and 2) Determining the diffusion coefficient based on the local curvature. Compared to the original model, the two modified cases could improve the Hack's law. However, the improvements for the exceedance probability of drainage area, discharge and upstream length were not impressive.

5.2 Outlook

Extension of large-scale concepts into unchanneled surfaces can lead to more accurate erosion and fluid flow estimations by providing simple statistical and physical metrics for surface evolution. To further enhance the applicability of the catchment-scale theories for the local soil management within the catchments, in the following, some experimental setups and modeling approaches are suggested for future works.

In a recent work, *Sassolas-Serrayet et al.* [2018] showed that the catchment's overall shape is more a function of Hack's coefficient rather than the exponent. Therefore, studying the scaling coefficients is as crucial as the exponents. On unchanneled surfaces with two slopes of 5 % and 10 %, we showed temporally invariant scaling laws with exponents in the range of catchment scales. The next step would be to find out

Chapter 5. Conclusions and outlook

how scaling coefficients and exponents are classified in that range. In our experiment, the surface water flow was accumulated at an outlet at the center of the flume's width. Changing the outlet position to have a non-symmetric condition is one way to create networks with different overall structures. It would be interesting to examine the effect of flume outlet on scaling exponents and coefficients and link them to the network orientation and density. Two other factors that influence networks' shapes are the flume main-slope and width to length ratio. We postulate that all networks will show the invariant scaling laws. However, the scaling exponents and coefficients are expected to have meaningful differences for different experimental scenarios. Having a broader laboratory studies helps a detailed clustering of the scaling laws.

In this dissertation, for the first time, an advective-diffusive LEM was calibrated with the experimental data. By calibration, we mean starting with the initial experimental morphology and conducting the numerical simulation of the whole course of the experiment with both advective and diffusive parts, and not estimating the parameters of the steady-state model based on the final state of the landscape. In contrast to the process-based models, a simplified flow field is assumed in the standard large-scale LEM. Therefore, it cannot render the flow velocity and the details within the water layer (suspended sediment particle distribution, the velocity, etc.). However, it showed to be a strong and simple method when the geomorphological patterns are the objective. For future studies, we highly recommend calibration of the model with experiments that include uplift and incised morphologies would. In order to test all aspects of the model, it should be calibrated based on both the elevation field and statistical feature of the rill network. In this work, the objective was to use the most typical form of the catchment-scale model and therefore, the D8 algorithm was used to extract the network. Examining the effect of the multi-directional algorithms of network extraction [Tarboton, 1997; Costa-Cabral and Burges, 1994; Freeman, 1991; Quinn *et al.*, 1991] on the statistical perspective of unchanneled morphologies would be another suggestion. Additionally, by conducting a similar experiment, comparing the application of the catchment-scale LEM with the process-based erosion models is also a valuable path to follow. In this regard, the objective would be to investigate the strengths and drawbacks of each model regarding the precision, calibration simplicity, uncertainty, and numerical simulation cost of the models.

Data availability statement

- The modeling, optimization and data analysis tools (C++ and Python codes) with a User's Guide are openly available at:

<https://github.com/mcheraghi/Landscape-Evolution-Model.git>.

- The surface morphology data are available at: <http://doi.org/10.5281/zenodo.1292113>.

A More details of the experimental setups

A.1 Surface scanning and image registration

In order to obtain the DEM data, Konica-Minolta VIVID 910 3D laser scanner was used. The scanner was successfully used before [Perona *et al.*, 2012; Edmaier *et al.*, 2015]. The device is capable of measuring the surface morphology with a precision of 10 μm . To do the scanning, the scanner was placed on a frame. The distance of the device from the flume surface was adjustable and the frame could move along the flume via rollers. Besides, it was possible to replace the scanner along the flume's width.

The device includes three different lens types: WIDE, MIDDLE and TELE. The first step in the preparation of the scanner was to test different lenses and distances to find out the optimum condition for the soil surface scanning. The dimension of the scanned piece and precision in the x and y directions are listed in Table A.1. A deflection test was performed by putting two completely straight aluminum bars on the sides of the flume. The deflection tests were done with 120 cm-WIDE and 180 cm-WIDE cases. Finally, the 120 cm-WIDE case was found to be the best because of easier replacement of the scanner on the frame, fewer oscillations of the frame and more precision in the registration and translation of different pieces.

Having the 28 pieces on the 6 m \times 2-m flume (Figure A.1), the areas ($a_1, a_2 \dots c_2$) and the points (A,..., D) were used for registration and coordination, respectively. An example of the scanned surface by using the above procedure is shown in Figure A.2.

Appendix A. More details of the experimental setups

Table A.1 – Dimension and resolution of the scanner with different lens types at various distances from the surface.

DISTANCE-LENS	width (mm)	Length (mm)	Δx (mm)	Δy (mm)
60 cm-TELE	120.94	90.99	0.38	0.38
60 cm-MIDDLE	212.95	160.69	0.65	0.69
60 cm-WIDE	380.45	285.25	1.18	1.23
120 cm-TELE	235.75	177.48	0.74	0.75
120 cm-MIDDLE	418.04	311.78	1.32	1.3
120 cm-WIDE	742.7	561.8	2.37	2.23
180 cm-TELE	346.31	260.8	1.09	1.09
180 cm-MIDDLE	609.73	457.34	1.93	1.92
180 cm-WIDE	1100.56	790.16	3.48	3.54

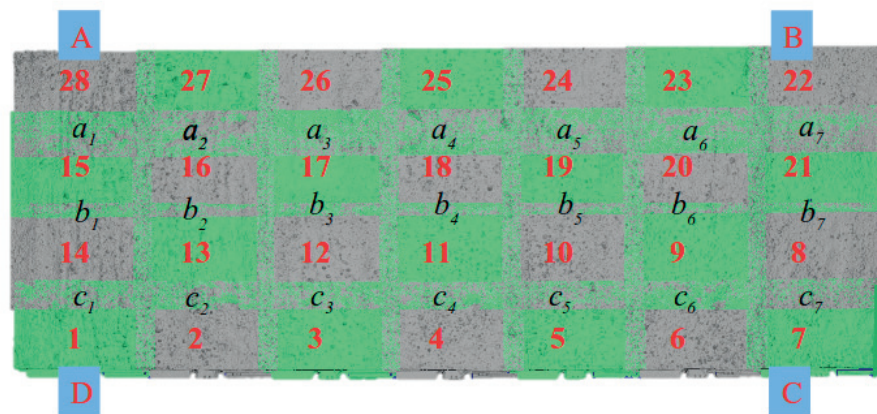


Figure A.1 – The pieces of scans on the flume. The numbers (1-28) show the order of scanning. The pieces were registered using the intersection areas ($a_1, a_2 \dots c_2$). Four points A, B, C, D were the labels on the flume's frame to coordinate the overall geometry.

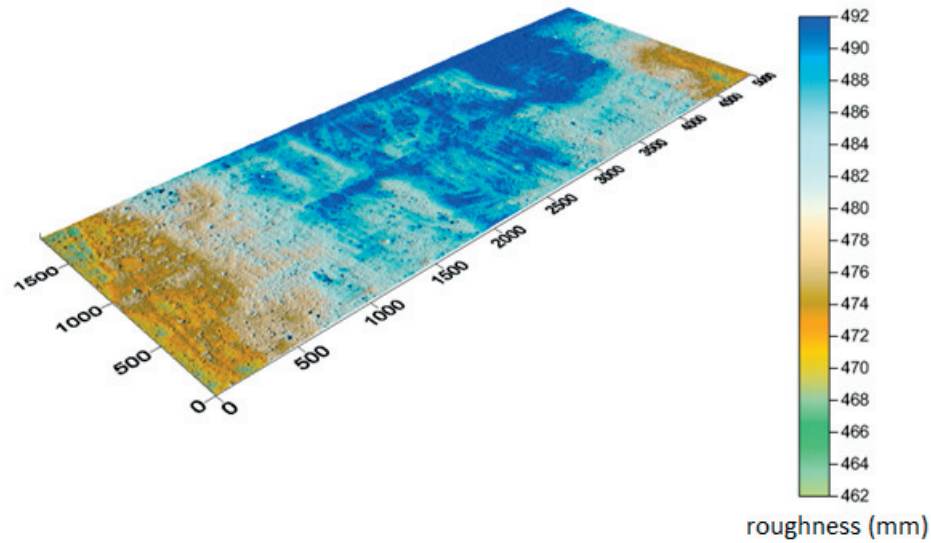


Figure A.2 – Digital Elevation Model of the flume. The 28 pieces of the 120cm-WIDE case were joined by translation along the width and registration along the flume's length. All values are in (mm).

A.2 Rainfall distribution measurement

In order to have justified calibrated parameters, it is much more efficient to put the precise rainfall distribution in the model. For obtaining a continuous profile of the rainfall intensity, the bottles were arranged throughout the flume as shown in Figure A.3. Afterward, the rainfall was generated for 30 minutes and the volumes of all the bottles were measured. This process was done for two slopes (2% and 10%). The experimental data were interpolated by modified Shepard method [*Franke and Nielson*, 1980] to make a grid in the size of the morphological cells.

A.3 Sediment selection for the experiment

We searched for an experiment which created an unchanneled morphology while there is a fast evolution. To this end, different sediment types, slope and rainfall intensities were tested (Table A.2). Experiment 1 was done on the large flume (6 m × 2 m, Figure 2.1) and the agricultural soil was initially compacted. After analyzing the DEMs, the variation on the surface was not enough. Therefore, we used an initially plowed dry soil in experiment 2. In this experiment, although initially some big rills

Appendix A. More details of the experimental setups



Figure A.3 – Rainfall distribution measurement on the flume.

A.3. Sediment selection for the experiment

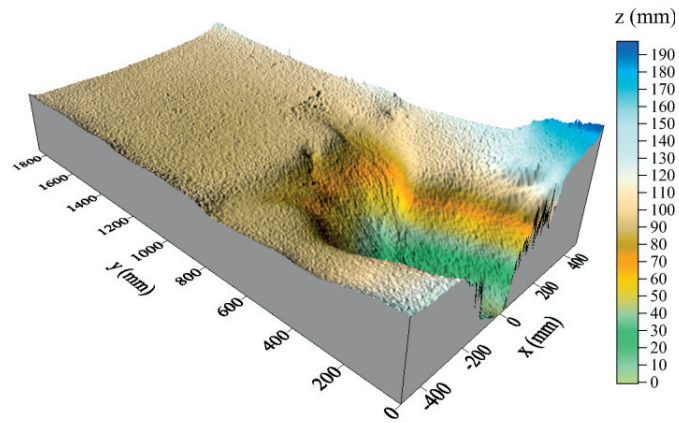
Table A.2 – The experiments carried out to achieve an unchanneled fast evolving morphology. Different rainfall intensities, sediment types and flume sizes were used.

Exp. No.	Sediment Type	Flume Size	Slope	Ave. R. I.	Duration	Surface Evolution
1	Cohesive	6 m × 2 m	10 %	60 mm h ⁻¹	2 h	✗
2	Cohesive	6 m × 2 m	10 %	60 mm h ⁻¹	2 h	✗
3	Cohesive	2 m × 1 m	10 %	80 mm h ⁻¹	2 h	✗
4	Uncohesive (coarse)	2 m × 1 m	10 %	80 mm h ⁻¹	4 h	✗
5	Uncohesive (fine)	2 m × 1 m	10 %	60 mm h ⁻¹	4 h	✓
6	Uncohesive (fine)	2 m × 1 m	3 %	80 mm h ⁻¹	20 h	✓
7	Uncohesive (fine)	2 m × 1 m	5 %	85 mm h ⁻¹	12 h	✓
8	Uncohesive (fine)	2 m × 1 m	5 %	85 mm h ⁻¹	16 h	✓

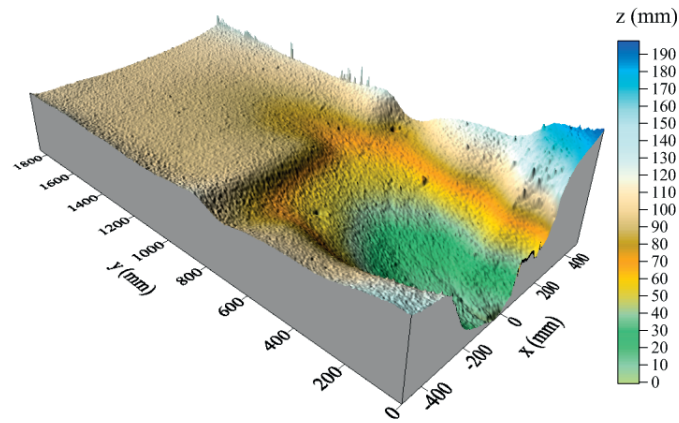
were formed on the flume's surface. But, they were disappeared after less than one hour due to deposition of sediments at the outlet. So, two changes were necessary: I) the flume outlet should not be open along the entire width and II) the initial sediment surface level should be higher than the outlet's surface.

In the next step, we made a smaller flume (2 m × 1 m) on which the flow was aggregated at the outlet and the soil surface was 15 cm above the outlet level (Figure3.1). Afterward, the agricultural soil was tested on the new flume in experiment 3 which was not successful. Finally, we decided to change the sediment on the small flume. In the experiments 4-6 the less cohesive sediment was tested. The surface evolution was observed for the fine lake sediments in the first few hours of experiments 5-8. The results of experiment 6 are shown in Figure A.4. Experiments 5 and 8 were used for the analysis in Chapters 3). The LEM was calibrated with the results of experiment 8 in Chapter 4). The surface morphology data are available at <http://doi.org/10.5281/zenodo.1292113>.

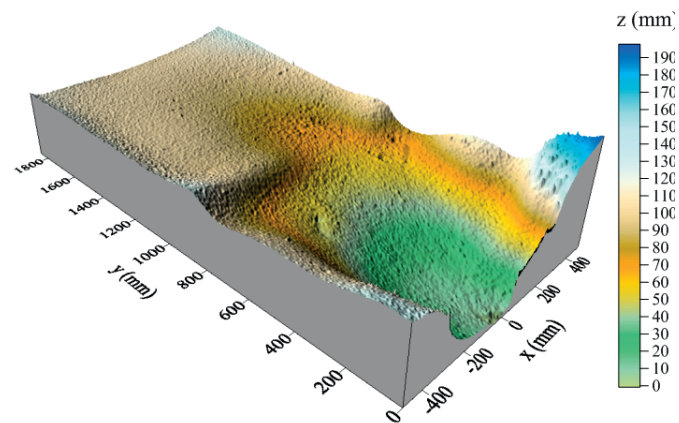
Appendix A. More details of the experimental setups



(a) $t = 1$ h



(b) $t = 5$ h



(c) $t = 9$ h

Figure A.4 – Morphology evolution of experiment 6 (Table A.2).

B Model verification

In this appendix, the C++ codes used for the numerical modeling of the morphology evolution is verified. The details of the numerical approach are presented in section 4.2. Two analytical solutions were used. In the first step, a 1-D advective-diffusive equation with periodic boundary condition will be solved. The second step, a Poisson equation was considered. Afterward, the model was tested to see the effect of diffusion on the incision rate.

B.1 1-D advection-diffusion

The 1-D advection-diffusion equation for $f(x, y)$ is in the form of:

$$\frac{\partial f}{\partial t} = D \frac{\partial^2 f}{\partial x^2} - K \frac{\partial f}{\partial x} \quad 0 \leq x \leq 2 \quad (\text{B.1})$$

with:

$$\begin{aligned} f(x, 0) &= A \sin(2\alpha\pi x) \\ f(0, t) &= f(2, t) \end{aligned} \quad (\text{B.2})$$

The analytical solution for this case is:

$$f(x, t) = e^{D\alpha^2 t} \sin(2\alpha\pi(x - Kt)) \quad (\text{B.3})$$

Considering, $D = 1$, $K = 1$, $A = 1$ and $\alpha = 1$, the numerical and analytical results are compared in Figure B.1. The numerical solution can capture the movement of the

Appendix B. Model verification

wave along x while the wave amplitude is reduced by time (t).

B.2 2-D diffusion

Diffusion equation for the function for $f(x, y, t)$ is:

$$\frac{\partial f}{\partial t} = \nabla^2 f + \psi \quad -1 \leq x \leq 1, \quad -1 \leq y \leq 1 \quad (\text{B.4})$$

where,

$$\begin{aligned} f(x, y, 0) &= 0 \\ f(-1, y, t) &= f(1, y, t) = f(x, -1, t) = f(x, 1, t) = 0 \end{aligned} \quad (\text{B.5})$$

The steady-state analytical solution ($-\nabla^2 f = \psi$) is:

$$f(x, y, +\infty) = \psi \left\{ \frac{1-x^2}{2} - \frac{16}{\pi^3} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\infty} \left\{ \frac{\sin(k\pi(1+x)/2)}{k^3 \sinh(k\pi)} \times (\sinh(k\pi(1+y)/2) + \sinh(k\pi(1-y)/2)) \right\} \right\} \quad (\text{B.6})$$

The numerical results of the initial times and the steady state solution are compared with the analytical results in Figure B.2 (e and f). As can be seen, the model and analytical results are in a very good agreement.

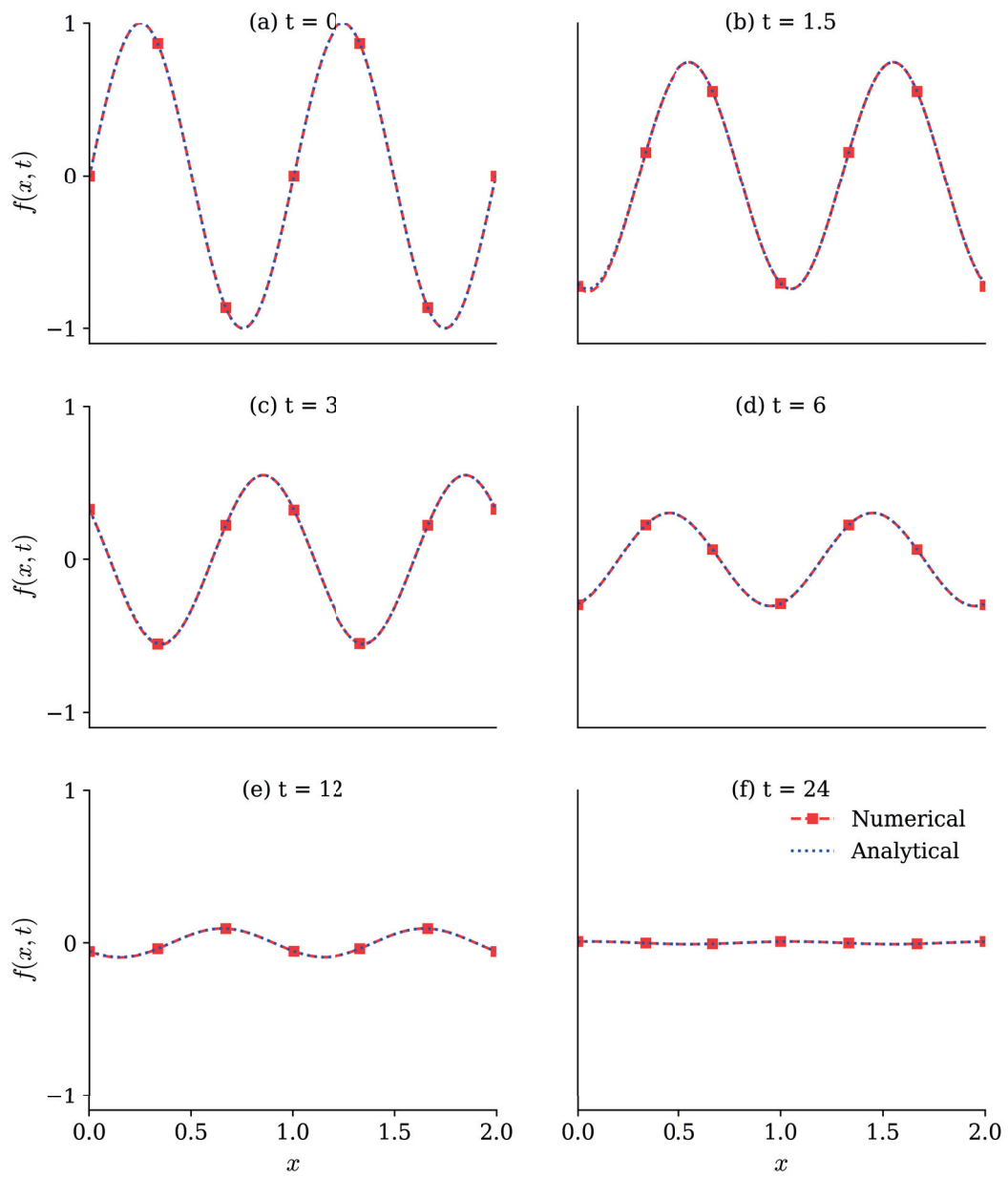


Figure B.1 – Analytical and numerical solution of equation B.1 with $D, K, A, \alpha = 1$.

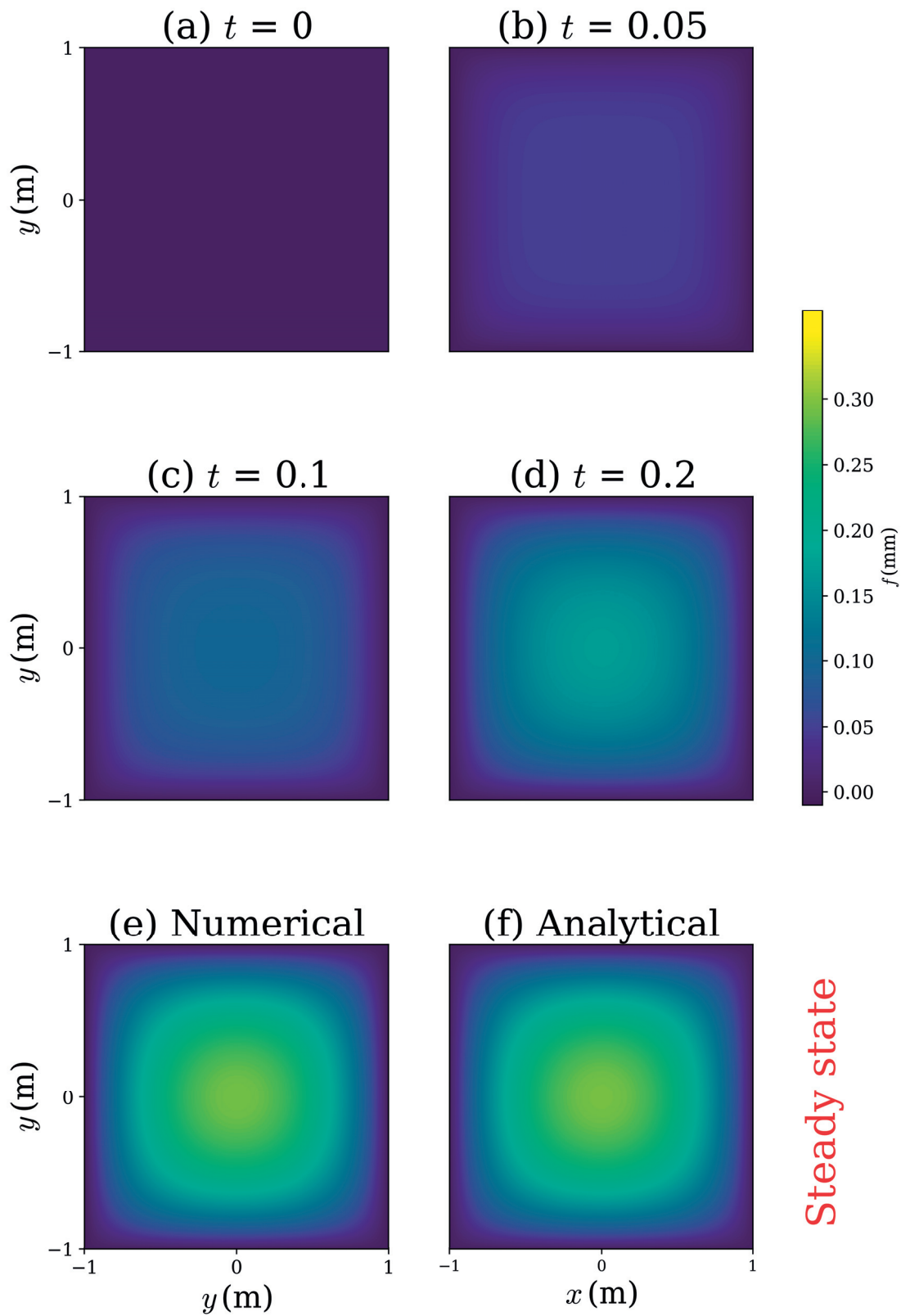


Figure B.2 – Numerical and analytical results of equation B.4 with $\psi = 1$: (a-d) Numerical results at the initial times, (e) numerical and (f) analytical (equation B.6) steady state solutions.

C C++ scripts: Landscape evolution model and optimization tools

The files with a User's Guide are available at:

<https://github.com/mcheraghi/Landscape-Evolution-Model.git>.

Listing C.1 – ‘header.cpp’: List of functions and their description

```
1 #include<iostream>
  #include<iomanip>
  #include<fstream>
  #include <string>
  #include <sstream>
  #include <cstdlib>
  #include <cmath>
  #include <deque>
  #include <ctime>
  #include <cstdlib>
11 #include <list>
  #include <sstream>
  #include <unistd.h>
  #include <chrono>
  #include <random>
  #include <thread>
  #include <tuple>
  using namespace std;
  using namespace std::this_thread; // sleep_for, sleep_until
  using namespace std::chrono; // nanoseconds, system_clock, seconds
21
  #define M 121//exp8: 241(4mm) , 121(8mm)    exp 7:123 (8 mm)//244 (4 mm)
  #define N 239 //exp8: 477(4mm) , 239(8mm)    exp 7:233 (8 mm)//464 (4 mm)
  #define dx 8
  #define dy dx
  #define delta dx
  #define mainSlope 0.05
  #define aveRain 84.0 //(mm/h)
  #define CFL 0.9
  #define Tmax 16.0
31 #define PI 3.14159265359

  struct objectives
  {
    double a; double b; double c; double d;
  };

  class interface
  {
```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
public:
41 int runOption;
void getRunOption(string addressRUN, string addressOPT);
};

class matrixOperations
{
public:
int transpose(double *A, double *B, int row, int col);
int multiple(double *A, double *B, double *C, int Ar, int Ac, int Br, int Bc);
int getMinor(double **src, double **dest, int row, int col, int order);
51 double calcDeterminant( double **mat, int order);
void matrixInversion(double *A, int order, double *Y);
friend class functionSet;
};

class functionSet:public matrixOperations
{
private:
int *DIR, *INPUT, *drainTo, *drainFrom;
double *CURVE, *R, *D, *alpha, *down, *up, *right, *left,*downBC, *upBC, *rightBC, *leftBC, *scanTime, *param,*Omega,*K;
61 double *a, *b, *c, *d;
ifstream input1,input2;
ofstream output1,output2, output3;

int PHICalc(double *Q, double *SLOPE, double *PHI);
int Thomas(double *,double *,double *,double *,double *,int);
void d2s(double num, char s[], unsigned int width);
public:
double E, Dave, Kave, m, n, OmegaAve,noiseAve;//E:tectonic, Dave:Diffusion, K:convection, m:discharge exponent,n:slope exponent
,Tave:critical stream power (Omega =Tave*random number) , noiseAve:coefficient of deposition term
71 double t;
int BCstep;
int varD;
double slopeLocal[8], dt, error;
double epsilon, gamma; //epsilon: the value for filling process, gamma: the value for BC
int allocate(double *X,double *Y,double *Z,double *Q,double *dZstar,double *Zstar,double *PHI);
void readLand(double *X,double *Y,double *Z, string str1);

int filling(double *, double *, double *);
void smoothZ(double *);
81 void direction(double *, double *, double * );
void L_down(double *);
void drainageQ(double *Q);
int mainRiver(double *X, double *Y, double *Z, double *L, double *BV, double BVmin, double *SLOPE, double *Q,
double *CURVE,double *LL, double *PHI, string folder, string fileName);//the main path is determined base on the
based value (BV) and its minimum value (BVmin)
int calcDrainFrom(double *BV);
void UpLenght(double *L);
int statistics(double *X, double *Y, double *Z, double *Q, double *AREA, double *SLOPE, double *CURVE, double *PHI,
double *L, double *LL, double percentage, string folder);
91 void prepareFile(string folderName, string fileName);
void drainageAREAandLL(double *AREA, double *LL, double *order); //The drainage area without considering the rainfall
void writeExceedance(double *paramMain, double *paramTemp, int NumPoints, string folderName, string fileName);
//it writes the exceedance probabilities of a specific parameter
void Kmean(double *X, double *Y, double *center, int numpoints, int myK, string folderName, string fileName);
void Ktest(double *X, double *Y, double *center, int maxK, string folderName, string fileName);
void findMinMax(double*param, double *min, double *max);
void normalize(double *X);
double average(double *X);
double RMS(double *X, int size);
101 double correlation(double *X, double *Y, int size);
double meanSquareError(double *X, double *Y, int size);
int Qcomponents(double *X, double *Y, double *Q, double *Qx, double *Qy);

void defineRain(string str1, int uni);
void defineRandom_Noise_D_K_Omega(double *Z,int varNoise, int varD,int varK,int varOmega);
void defineD(int varD);
void defineRandomD(int varD);// 0 or 1 is put directly in the function
void defineK(int varK);
void defineOmega(int varOmega);
```

```

111 void slopeCalc(double *Z, double *SLOPE);
void curveCalc(double *SLOPE, double *CURVE);
objectives multiObjective_Zerror_discharge(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE,
double *PHI, double *dZstar, string initialTime, string finalTime, string* filename);
double objective_discharge(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI,
double *dZstar, string initialTime, string finalTime, string* filename);
int temporalBCset(double *X, double *Y, double *Z, string* filename); //sets the boundary according to the experimental data
void filter33(double *param, double *filtered, int filtTime);
double filternn(double *param, double *filtered, int n); //it filters param by an n by n cells
double filterCELL(double *param, int i, int j, int n); //it filters param[i,j] by an n by n cells
121 void filterNetwork(double *Z1, double *Z2, double *Y, double *Qopt);
int dtCalc(double *Q, double *SLOPE);
int RungeKutta(double *Z, double *Zstar, double *Q, double *SLOPE, double *dZstar, double *PHI, double *Y);
int crankNicholson(double *Zstar, double *Y);
int boundarySet(double *Z, double *Y);
int replaceError(double *Z, double *Zstar);
int addNoise(double *Z, double dt); //in numeric file
int writeDATA(double *X, double *Y, double *Z, double *X1, double *X2, double *X3, double *X4, double *X5, double *X6,
double *X7, string folder, int step);
int writeXYZ(double *X, double *Y, double *Z, string folder, int step);
131 int analytic(double *X, double *Y, double *Z);
int RUN(double *X, double *Y, double *Z, double *Zstar, double *Q, double *Qx, double *Qy, double *SLOPE, double *CURVE,
double *PHI, double *dZstar, double *AREA, double *LL, double *downL, string addressRUN, double initialTime,
double finalTime, string initialTimeZ, string finalTimeZ);
double objective1(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI, double *dZstar,
string initialTime, string finalTime, string* filename);
double stepRun(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI, double *dZstar,
double initialT, double finalT, string finalZ);
double objective2(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI, double *dZstar);
int calibratePSO(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI, double *dZstar,
141 string addressOPT, string initialTime, string finalTime, string* filename);
int calibrateMontCarlo(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI);
int confidenceInterval(double q[], double percentage[], double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE,
double *PHI, double *dZstar, double T1, double T2, string initialTime, string finalTime);
int confidenceInterval1(double q0[], double percentage[], double *X, double *Y, double *Z, double *Zstar, double *Q,
double *SLOPE, double *PHI, double *dZstar, double T1, double T2, string initialTime, string finalTime);
void SetCursorPos(int XPos, int YPos);
};

```

Listing C.2 – ‘main.cpp’: The main functions

```

#include "header.h"
//The output folder, the input DEM file and the rain contour has to be specified,
void mainRUN()
{
double *X,*Y,*Z,*Q, *Qx, *Qy, *AREA,*SLOPE,*CURVE,*L,*LL,*downL,*Zstar,*dZstar,*PHI,*order;
// the parameter dzstar in mainRUN is replaced by L
int RunOpt, i, j;

X      = (double*) calloc(M*N , sizeof(double));
10 Y      = (double*) calloc(M*N , sizeof(double));
Z      = (double*) calloc(M*N , sizeof(double));
Q      = (double*) calloc(M*N , sizeof(double));
Qx     = (double*) calloc(M*N , sizeof(double));
Qy     = (double*) calloc(M*N , sizeof(double));
SLOPE  = (double*) calloc(M*N , sizeof(double));
CURVE  = (double*) calloc(M*N , sizeof(double));
Zstar  = (double*) calloc(M*N , sizeof(double));
dZstar = (double*) calloc(M*N , sizeof(double));
PHI     = (double*) calloc(M*N , sizeof(double));
20 L     = (double*) calloc(M*N , sizeof(double));
AREA   = (double*) calloc(M*N , sizeof(double));
LL     = (double*) calloc(M*N , sizeof(double));
downL  = (double*) calloc(M*N , sizeof(double));
order  = (double*) calloc(M*N , sizeof(double));
interface III;
functionSet MCH;
MCH.allocate(X,Y,Z,Q,dZstar, Zstar,PHI);
MCH.Dave =17571.0;
MCH.K = 0.185;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
30 MCH.m = 0.5;
MCH.n = 1.0;
MCH.tetaC = 13.07;
MCH.E = 0;
MCH.rndCoef = 10;
char *path1=NULL;
    size_t size;
    string path=getcwd(path1,size);
    cout<<"\n current Path"<<path;
    path = path.substr(0,path.size()-6);    // take the "/model" part
40 string addressRUN = path+"/results";
    //boost::filesystem::path full_path( boost::filesystem::current_path() );
    //full_path-="/model";
    //string addressRUN = "/home/mohsen/Man2/DATA/RUNS/run1/results"; //the output file
MCH.varD = 0; //0: uniform, 1: variable D
double initialTime = 0.25; //NOT 0.5 and 1.0!
double finalTime = 16.0; //NOT 0.5 and 1.0!
string initialTimeZ, finalTimeZ;
if (dx == 8)
{
50 if (initialTime == 0.25)
    initialTimeZ = path+"/input/00h15min_8mm.dat"; //the input file
    else if (initialTime == 0.5)
    initialTimeZ = path+"/input/00h30min_8mm.dat"; //the input file
    else if (initialTime == 1.0)
    initialTimeZ = path+"/input/01h00min_8mm.dat"; //the input file
    else if (initialTime == 2.0)
    initialTimeZ = path+"/input/02h00min_8mm.dat"; //the input file
    else if (initialTime == 4.0)
    initialTimeZ = path+"/input/04h00min_8mm.dat"; //the input file
60 else if (initialTime == 8.0)
    initialTimeZ = path+"/input/08h00min_8mm.dat"; //the input file
    if (finalTime == 0.25)
    finalTimeZ = path+"/input/00h15min_8mm.dat"; //the input file
    else if (finalTime == 2.0)
    finalTimeZ = path+"/input/02h00min_8mm.dat"; //the input file
    else if (finalTime == 4.0)
    finalTimeZ = path+"/input/04h00min_8mm.dat"; //the input file
    else if (finalTime == 8.0)
    finalTimeZ = path+"/input/08h00min_8mm.dat"; //the input file
70 else if (finalTime == 16.0)
    finalTimeZ = path+"/input/16h00min_8mm.dat"; //the input file

    MCH.defineRain (path+"/input/Rain8mm.dat",1);    /// const = 0:uniform rain
}
else
{
    if (initialTime == 0.25)
    initialTimeZ = "/home/mohsen/LandScapeCode/input/00h15min_4mm.dat"; //the input file
    if (initialTime == 0.5)
80 initialTimeZ = "/home/mohsen/LandScapeCode/input/00h30min_4mm.dat"; //the input file
    if (initialTime == 1.0)
    initialTimeZ = "/home/mohsen/LandScapeCode/input/01h00min_4mm.dat"; //the input file
    else if (initialTime == 2.0)
    initialTimeZ = "/home/mohsen/LandScapeCode/input/02h00min_4mm.dat"; //the input file
    else if (initialTime == 4.0)
    initialTimeZ = "/home/mohsen/LandScapeCode/input/04h00min_4mm.dat"; //the input file
    else if (initialTime == 8.0)
    initialTimeZ = "/home/mohsen/LandScapeCode/input/08h00min_4mm.dat"; //the input file
    if (finalTime == 0.25)
90 finalTimeZ = "/home/mohsen/LandScapeCode/input/00h15min_4mm.dat"; //the input file
    else if (finalTime == 2.0)
    finalTimeZ = "/home/mohsen/LandScapeCode/input/02h00min_4mm.dat"; //the input file
    else if (finalTime == 4.0)
    finalTimeZ = "/home/mohsen/LandScapeCode/input/04h00min_4mm.dat"; //the input file
    else if (finalTime == 8.0)
    finalTimeZ = "/home/mohsen/LandScapeCode/input/08h00min_4mm.dat"; //the input file
    else if (finalTime == 16.0)
    finalTimeZ = "/home/mohsen/LandScapeCode/input/16h00min_4mm.dat"; //the input file

100 MCH.defineRain ("/home/mohsen/LandScapeCode/input/Rain4mm.dat",0);    ///
```

```

}
string filename[] = {path+"/input/00h15min_8mm.dat", path+"/input/00h30min_8mm.dat", path+"/input/01h00min_8mm.dat",
path+"/input/02h00min_8mm.dat", path+"/input/04h00min_8mm.dat", path+"/input/08h00min_8mm.dat", path+"/input/16h00min_8mm.dat"};

if (dx == 4) //attention that the file names have the same size
{
    string filename[] = {" /home/mohsen/LandScapeCode/input/00h15min_4mm.dat", " /home/mohsen/LandScapeCode/input/00h30min_4mm.dat",
    " /home/mohsen/LandScapeCode/input/01h00min_4mm.dat", " /home/mohsen/LandScapeCode/input/02h00min_4mm.dat",
    " /home/mohsen/LandScapeCode/input/04h00min_4mm.dat", " /home/mohsen/LandScapeCode/input/08h00min_4mm.dat",
110 " /home/mohsen/LandScapeCode/input/16h00min_4mm.dat"};
    cout <<filename[1] << endl;
}
MCH.temporalBCset(X, Y, Z,filename); // captures the BCs from the scans
MCH.RUN(X, Y, Z, Zstar, Q, Qx, Qy, SLOPE, CURVE, PHI, dZstar, AREA, LL,downL, addressRUN, initialTime, finalTime,
initialTimeZ, finalTimeZ); //0:constant D, 1: variable D [refer to defined(int varD) function]
//MCH.mainRiver(X, Y, Z, Zstar, Q, SLOPE, CURVE, PHI, dZstar, addressRUN);
}

120 void mainOPT()
{
    time_t tstart, tend;
    tstart = time(0);
    double *X,*Y,*Z,*Q,*SLOPE,*CURVE,*dZstar,*Zstar,*PHI;
    X = (double*) calloc(M*N, sizeof(double));
    Y = (double*) calloc(M*N, sizeof(double));
    Z = (double*) calloc(M*N, sizeof(double));
    Q = (double*) calloc(M*N, sizeof(double));
    SLOPE = (double*) calloc(M*N, sizeof(double));
    CURVE = (double*) calloc(M*N, sizeof(double));
130 dZstar = (double*) calloc(M*N, sizeof(double));
    Zstar = (double*) calloc(M*N, sizeof(double));
    PHI = (double*) calloc(M*N, sizeof(double));
    interface III;
    functionSet MCH;
    double T1 = 0.25;
    double T2 = 8;
    char *path1=NULL;
    size_t size;
140 string path=getcwd(path1,size);
    cout<<"\n current Path"<<path<<endl;
    path = path.substr(0,path.size()-6); // take the "/model" part
    string addressOPT = path+"/results";
    MCH.allocate(X,Y,Z,Q,dZstar, Zstar,PHI);
    // string addressOPT = "/home/mohsen/LandScapeCode/2D/Results/exp8/OPTIMIZATION/interpolatedBC/confidenseInterval/uniD_opAt8h.dat";
    //the output file for optimization
    MCH.varD = 0;
    string initialTime = path+"/input/00h15min_8mm.dat"; //the input file
    string finalTime = path+"/input/08h00min_8mm.dat"; //the input file
150 MCH.defineRain(path+"/input/Rain8mm.dat",1); ///

    string filename[] = {path+"/input/00h15min_8mm.dat", path+"/input/00h30min_8mm.dat", path+"/input/01h00min_8mm.dat",
    path+"/input/02h00min_8mm.dat", path+"/input/04h00min_8mm.dat", path+"/input/08h00min_8mm.dat", path+"/input/16h00min_8mm.dat"};

    if (dx == 4) //attention that the file names have the same size
    {
        string filename[] = {" /home/mohsen/LandScapeCode/input/00h15min_4mm.dat", " /home/mohsen/LandScapeCode/input/00h30min_4mm.dat",
        " /home/mohsen/LandScapeCode/input/01h00min_4mm.dat", " /home/mohsen/LandScapeCode/input/02h00min_4mm.dat",
        " /home/mohsen/LandScapeCode/input/04h00min_4mm.dat", " /home/mohsen/LandScapeCode/input/08h00min_4mm.dat",
160 " /home/mohsen/LandScapeCode/input/16h00min_4mm.dat"};
        cout <<filename[1] << endl;
    }

    MCH.temporalBCset(X, Y, Z,filename); // captures the BCs from the scans
    MCH.calibratePSO(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, addressOPT, initialTime,finalTime,filename);
    double q[3] = {17571,0.184997,13.0765};
    double percentage[3] = {0.1,0.1,0.1};
    tend = time(0);
    cout << "It took "<< difftime(tend, tstart)/3600 <<" hours"<< endl;
170 }

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
void mainAnalysis ()
{
    double *X,*Y,*Z,*Zmodel,*Q, *Qx, *Qy, *AREA,*SLOPE,*CURVE,*downL,*L,*LL,*Zstar,*PHI, *power, *SLOPE_CURVE, *order;
    // the parameter dzstar in mainRUN is replaced by L
    int RunOpt, i, j;
    X = (double*) calloc (M*N , sizeof(double));
    Y = (double*) calloc (M*N , sizeof(double));
180 Z = (double*) calloc (M*N , sizeof(double));
    Zmodel = (double*) calloc (M*N , sizeof(double));
    Q = (double*) calloc (M*N , sizeof(double));
    Qx = (double*) calloc (M*N , sizeof(double));
    Qy = (double*) calloc (M*N , sizeof(double));
    SLOPE = (double*) calloc (M*N , sizeof(double));
    CURVE = (double*) calloc (M*N , sizeof(double));
    SLOPE_CURVE = (double*) calloc (M*N , sizeof(double));
    Zstar = (double*) calloc (M*N , sizeof(double));
    PHI = (double*) calloc (M*N , sizeof(double));
190 L = (double*) calloc (M*N , sizeof(double));
    AREA = (double*) calloc (M*N , sizeof(double));
    LL = (double*) calloc (M*N , sizeof(double));
    downL = (double*) calloc (M*N , sizeof(double));
    power = (double*) calloc (M*N , sizeof(double));
    order = (double*) calloc (M*N , sizeof(double));

    interface III;
    functionSet MCH;
    MCH.allocate (X,Y,Z,Q,L, Zstar,PHI);
200 MCH.epsilon = 0.05;
    string r = "0.5";
    char *path1=NULL;
    size_t size;
    string path=getcwd(path1,size);
    cout<<"\n current Path"<<path;
    path = path.substr(0,path.size()-6); // take the "/model" part
    string folder = path+"/results";
    string fileName[] = { path+"/input/00h15min_8mm.dat",
        path+"/input/00h30min_8mm.dat",
210 path+"/input/01h00min_8mm.dat",
        path+"/input/02h00min_8mm.dat",
        path+"/input/04h00min_8mm.dat",
        path+"/input/08h00min_8mm.dat",
        path+"/input/16h00min_8mm.dat" };
    MCH.defineRain(path+"/input/Rain8mm.dat",0); //
    string modelName[] = {" /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ00..dat",
        " /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ01..dat",
        " /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ02..dat",
        " /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ03..dat",
220 " /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ04..dat",
        " /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ05..dat",
        " /home/mohsen/LandScapeCode/2D/ Results /exp8/RUNS/87_8mm/XYZ06..dat" };
    string outName[] = {"0.25h.csv", "0.5h.csv", "1h.csv", "2h.csv", "4h.csv", "8h.csv", "16h.csv"};
    int ii, jj;
    for (i = 0; i < 7; i++)
    {
        string input = fileName[i]; //the input file
        string makefolder = "mkdir " + folder;
        system(makefolder.c_str());
230 MCH.readLand(X, Y, Z,input);
        //MCH.filternn(Z,Zstar, 30);
        //MCH.replaceError(Z,Zstar); //Zstar to Z
        MCH.defineRain(" /home/mohsen/LandScapeCode/input/Rain8mm.dat",0);
        MCH.temporalBCset(X, Y, Zstar,fileName);
        MCH.filling(Z, PHI, Y);
        MCH.direction(Z, PHI, Y);

        MCH.drainageQ(Q);
        MCH.drainageAREAandLL(AREA, LL,order);
240 MCH.L_down(downL);
        MCH.slopeCalc(Z,SLOPE);
        MCH.curveCalc(Z,CURVE);
```



```

//MCH.Qcomponents(X, Y, Q, Qx,Qy);
MCH.writeDATA(X, Y,Z,Z,SLOPE,Q,AREA,LL, CURVE,downL, folder, i);
MCH.writeXYZ(X, Y,Z, folder, i);

}

double min, max;
MCH.findMinMax(AREA, &min, &max);
cout << min << " " << max << endl;
250 MCH.findMinMax(Q, &min, &max);
cout << min << " " << max << endl;
}

int main()
{
int mainNumb;

cout << " Bonjour Mohsen, Entre le nombre de fichier (RUN:1, OPT:2 and ANALYSIS: 3)" << endl;
cin >> mainNumb;
cout << " Attention: Le resolution est: " << dx << " mm" << endl;
cout << " Je vais checker" << endl;
if (mainNumb == 1)
{
cout << " OK, On y va :) !!!" << endl;
mainRUN();
}
else if (mainNumb == 2)
{
270 cout << " OK, On y va :) !!!" << endl;
mainOPT();
}
else if (mainNumb == 3)
{
cout << " OK, On y va :) !!!" << endl;
mainAnalysis();
}
else
{
280 cout << "NO, No, Le shifre doit etre 1, 2, ou 3!!!!" << endl;
}
return 0;
}

//for analytical solution, changes:dx dz delta,M,N, E = 1, K = 0, in PHI function if ( K==0 PHI = E),readinput function,
/*double ERROR = 5;
int num = 0;
double t = 0;
double step = Tmax/10.0, captureTime = t + step;
290 MCH.writeDATA(X, Y, Z, Q, num);
while (t < Tmax)
{
MCH.dtCalc(Q, SLOPE);
MCH.RungeKutta(Z, Zstar, Q, SLOPE, dZstar, PHI);
MCH.crankNicholson(Z, Zstar);
MCH.replaceError(Z, Zstar);
MCH.boundarySet(Z);
t+=MCH.dt;
ERROR = MCH.error;
300 cout << "CHANGE (MM) =" << ERROR << " " << t << endl;
if (t>captureTime)
{MCH.writeDATA(X, Y, Z, Q, ++num);captureTime+=step;}
}

MCH.analytic(X, Y, Z);*/

```

Listing C.3 – ‘network.cpp’: Drainage network extraction

```

#include "header.h"
int functionSet::allocate(double *X,double *Y,double *Z,double *Q,double *dZstar, double *Zstar,double *PHI)
{
int i,j;

```



```

        if(R[j*M+i] < 0)
            R[j*M+i] = 0;
        R[j*M+i] *= aveRain;          //for variable rainfall it is R[j*M+i] = *aveRain;
    }

    input1.close();
}
85 }

void functionSet::defineD(int varD)
{
    int i,j;
    if(varD == 0)
    {
        for(j=0; j<N; j++)
        for(i=0; i<M; i++)
95     D[j*M+i] = Dave; //D[j*M+i] = Dave*R[j*M+i]/aveRain; D[j*M+i] = Dave
    }
    else
    {
        for(j=0; j<N; j++)
        for(i=0; i<M; i++)
            D[j*M+i] = Dave*R[j*M+i]/aveRain; //D[j*M+i] = Dave*R[j*M+i]/aveRain; D[j*M+i] = Dave
    }
}

105 void functionSet::defineRandom_Noise_D_K_Omega(double *Z,int varNoise,int varD,int varK,int varOmega)
//it calculates the parameters according to the value and sign of the noises
{
    double random,rand1;
    double noise;
    int i,j;
    std::random_device rd;
    std::default_random_engine generator(rd());
    std::cauchy_distribution<double> cauchy(0,0.11757900431008157);
    for(i = 1; i<M-1; i++)
115     for(j = 1; j<N-1; j++) //adding the noise into the field
    {
        random = 10;
        while (random>1)
            random = abs(cauchy(generator));
        noise = noiseAve*random*dt;
        rand1 = (double) rand() / (RAND_MAX);
        if(rand1>0.5)
        {
            if(varNoise != 0)
125             Z[j*M + i] = Z[j*M + i] + noise;
            if(varD != 0)
                D[j*M+i] = Dave*(1-random);
            else
                D[j*M+i] = Dave;
            if(varK != 0)
                K[j*M+i] = Kave*(1-random);
            else
                K[j*M+i] = Kave;
            if(varOmega != 0)
135             Omega[j*M+i] = OmegaAve*random;
            else
                Omega[j*M+i] = OmegaAve;
        }

        else if(rand1<0.5)
        {
            if(varNoise != 0)
                Z[j*M + i] = Z[j*M + i] - noise;
            if(varD != 0)
145             D[j*M+i] = Dave*(1-random);
            else

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
D[j*M+i] = Dave;
if (varK != 0)
    K[j*M+i] = Kave*random;
else
    K[j*M+i] = Kave;
if (varOmega != 0)
    Omega[j*M+i] = OmegaAve*(1-random);
else
155   Omega[j*M+i] = OmegaAve;
}
}
}

void functionSet::defineRandomD(int varD) // 0 or 1 is put directly in the function
{
    srand ( time(NULL) );
    int i, j;
    double random;
165   std::random_device rd;
        std::default_random_engine generator(rd());
    std::cauchy_distribution<double> cauchy(0,0.11757900431008157);
    if (varD == 0)
    {
        for (j=0; j<N; j++)
        for (i=0; i<M; i++)
            D[j*M+i] = Dave; //D[j*M+i] = Dave*R[j*M+i]/aveRain;D[j*M+i] = Dave
    }
    else
175   {
        for (j=0; j<N; j++)
        for (i=0; i<M; i++)
        {
            random = 10;
            while (random>1)
                random = abs(cauchy(generator));

            D[j*M+i] = Dave*(1-random); //D[j*M+i] = Dave*R[j*M+i]/aveRain;D[j*M+i] = Dave
        }
185   }
}

void functionSet::defineK(int varK) //varK ==0:fixed K, varK ==1:random K
{
    int i, j;
    double random;
    std::random_device rd;
        std::default_random_engine generator(rd());
    std::cauchy_distribution<double> cauchy(0,0.11757900431008157);
195   if (varK == 0)
    {
        for (j=0; j<N; j++)
        for (i=0; i<M; i++)
            K[j*M+i] = Kave; //D[j*M+i] = Dave*R[j*M+i]/aveRain;D[j*M+i] = Dave
    }
    else
    {
        for (j=0; j<N; j++)
        for (i=0; i<M; i++)
205   {
            random = 10;
            while (random>1)
                random = abs(cauchy(generator));
            K[j*M+i] = Kave*(1-random); //D[j*M+i] = Dave*R[j*M+i]/aveRain;D[j*M+i] = Dave
        }
    }
}

215 void functionSet::defineOmega(int varOmega)
{
    int i, j;
```


Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
    if (Z[(N - 1)*M + i] <= Z[(N - 2)*M + i] + epsilon )
        Z[(N - 1)*M + i] = Z[(N - 2)*M + i] + epsilon;
}

295 Z[0*M + M/2] = 0;

for (j = 0; j < N ; j++)
{
    if (Z[j*M + 0] <= Z[j*M + 1] + epsilon )
        Z[j*M + 0] = Z[j*M + 1] - epsilon;

    if (Z[j*M + M - 1] <= Z[j*M + M - 2] + epsilon )
        Z[j*M + M - 1] = Z[j*M + M - 2] - epsilon;
305 }
Z[0*M + 0] = (Z[1*M + 0] + Z[0*M + 1]) / 2;
Z[0*M + M - 1] = (Z[1*M + M - 1] + Z[0*M + M - 2]) / 2;
Z[(N - 1)*M + 0] = (Z[(N - 1)*M + 1] + Z[(N - 2)*M + 0]) / 2;
Z[(N - 1)*M + M - 1] = (Z[(N - 2)*M + M - 1] + Z[(N - 1)*M + M - 2]) / 2;

for (j=1; j<N-1; j++)
{
    Zstar[j*M + 0] = Z[j*M + 0];
    Zstar[j*M + M - 1] = Z[j*M + M - 1];
315 }

for (i=1; i<M-1; i++)
{
    Zstar[0*M + i] = Z[0*M + i];
    Zstar[(N-1)*M + i] = Z[(N-1)*M + i];
}

for (j=N-2; j>0; j--)
for (i=1; i<M-1; i++)
325 {
    Zstar[j*M + i] = 20000;
}

for (j=1; j<N-1; j++)
for (i=1; i<M-1; i++)
{

    min = Z[j*M + i + 1];
    if (Z[(j - 1)*M + i + 1] < min)
335     min = Z[(j - 1)*M + i + 1];

    if (Z[(j - 1)*M + i] < min)
        min = Z[(j - 1)*M + i];

    if (Z[(j - 1)*M + i - 1] < min)
        min = Z[(j - 1)*M + i - 1];

    if (Z[(j - 0)*M + i - 1] < min)
        min = Z[(j - 0)*M + i - 1];
345

    if (Z[(j + 1)*M + i - 1] < min)
        min = Z[(j + 1)*M + i - 1];

    if (Z[(j + 1)*M + i - 0] < min)
        min = Z[(j + 1)*M + i - 0];

    if (Z[(j + 1)*M + i + 1] < min)
        min = Z[(j + 1)*M + i + 1];

355 if (Z[j*M + i] > min + epsilon)
    Zstar[j*M + i] = Z[j*M + i];
else if (Zstar[j*M + i] > min + epsilon )
{
    Zstar[j*M + i] = min + epsilon ;
}
```

```

        count++;
    }
}
for(j=0; j<N; j++)
for(i=0; i<M; i++)
365 Z[j*M + i] = Zstar[j*M + i];
for(j=0; j<N; j++)
for(i=0; i<M; i++)
    Zstar[j*M + i] = 0;
return 0;
}

void functionSet::direction(double *Z, double *Zstar, double *Y)
{
    int i, j;
375 double max;

    for(j=0; j<N; j++)
        for(i=0; i<M; i++)
            drainTo[j*M + i] = -1;
    for(j=1; j<N-1; j++)
    {
        DIR[j*M + 0] = 1;
        drainTo[j*M + 0] = j*M + 1;
        DIR[j*M + M - 1] = 16;
385 drainTo[j*M + M - 1] = j*M + M - 2;
    }

    for(i=1; i<M-1; i++)
    {
        DIR[0*M + i] = 64;
        drainTo[0*M + i] = 1*M + i;

        DIR[(N-1)*M + i] = 4;
        drainTo[(N-1)*M + i] = (N-2)*M + i;
395 i = M/2;
        DIR[0*M + i] = -1;
        drainTo[0*M + i] = -1;
        DIR[1*M + i] = 4;
        drainTo[1*M + i] = 0*M + i;

        double count = 10, count1 = 0;
        while (count>0)
        {
            int maxIndex;
405 for(j=1; j<N-1; j++)
            for(i=1; i<M-1; i++)
            {
                slopeLocal[0] = (Z[j*M + i] - Z[j*M + i + 1]);
                slopeLocal[1] = (Z[j*M + i] - Z[(j - 1)*M + i + 1])/1.4142;
                slopeLocal[2] = (Z[j*M + i] - Z[(j - 1)*M + i]);
                slopeLocal[3] = (Z[j*M + i] - Z[(j - 1)*M + i - 1])/1.4142;
                slopeLocal[4] = (Z[j*M + i] - Z[j*M + i - 1]);
                slopeLocal[5] = (Z[j*M + i] - Z[(j + 1)*M + i - 1])/1.4142;
                slopeLocal[6] = (Z[j*M + i] - Z[(j + 1)*M + i]);
415 slopeLocal[7] = (Z[j*M + i] - Z[(j + 1)*M + i + 1])/1.4142;
                maxIndex = 0;
                max = slopeLocal[0] ;
                int k;
                for (k = 1; k < 8; k++)
                    if (slopeLocal[k] > max)
                    {
                        max = slopeLocal[k];
                        maxIndex = k;
                    }
425 DIR[j*M + i] = -1;
                drainTo[j*M + i] = -1;

                if (maxIndex == 0 )
                {
                    drainTo[j*M + i] = j*M + i + 1;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
    DIR[j*M + i] = 1;
}
if (maxIndex == 1 )
{
435   drainTo[j*M + i] = (j - 1)*M + i + 1;
   DIR[j*M + i] = 2;
}
if (maxIndex == 2 )
{
   drainTo[j*M + i] = (j - 1)*M + i;
   DIR[j*M + i] = 4;
}
if (maxIndex == 3 )
{
445   drainTo[j*M + i] = (j - 1)*M + i - 1;
   DIR[j*M + i] = 8;
}
if (maxIndex == 4 )
{
   drainTo[j*M + i] = j*M + i - 1;
   DIR[j*M + i] = 16;
}
if (maxIndex == 5)
{
455   drainTo[j*M + i] = (j + 1)*M + i - 1;
   DIR[j*M + i] = 32;
}
if (maxIndex == 6)
{
   drainTo[j*M + i] = (j + 1)*M + i;
   DIR[j*M + i] = 64;
}
if (maxIndex == 7)
{
465   drainTo[j*M + i] = (j + 1)*M + i + 1;
   DIR[j*M + i] = 128;
}
}
countl = 0;
for(j=1; j<N-1; j++)
for(i=1; i<M-1; i++)
    if(drainTo[drainTo[j*M + i]] == j*M + i || drainTo[drainTo[drainTo[j*M + i]]] == j*M + i
    || drainTo[drainTo[drainTo[drainTo[j*M + i]]]] == j*M + i || drainTo[j*M + i] == -1)
        countl++;
475
if (countl > 0)
    filling(Z, Zstar, Y);
//cout << "PIT POIT: " << countl << endl;
count = countl;
}
//cout << "direction() is finished" << endl;
}

void functionSet::drainageQ(double *Q)
485 {
    int i, j;
    for(j=0; j<N; j++)
    {
        Q[j*M + 0] = R[j*M + 0] ;
        INPUT[j*M + 0] = 0;

        Q[j*M + M - 1] = R[j*M + M - 1];
        INPUT[j*M + M - 1] = 0;
    }
495    for(i=0; i<M; i++)
    {
        Q[0*M + i] = R[0*M + i] ;
        INPUT[0*M + i] = 0;

        Q[(N-1)*M + i] = R[(N-1)*M + i];
        INPUT[(N-1)*M + i] = 0;
    }
}
```

```

    }
    for(j=N-2; j>0; j--)
    for(i=1; i<M-1; i++)
505 {
        Q[j*M + i] = R[j*M + i];
        INPUT[j*M + i] = 0;

        if (DIR[j*M + i + 1] == 16)
            INPUT[j*M + i] += 1;

        if (DIR[(j + 1)*M + i + 1] == 8)
            INPUT[j*M + i] += 1;

515 if (DIR[(j + 1)*M + i] == 4)
            INPUT[j*M + i] += 1;

        if (DIR[(j + 1)*M + i - 1] == 2)
            INPUT[j*M + i] += 1;

        if (DIR[j*M + i - 1] == 1)
            INPUT[j*M + i] += 1;

        if (DIR[(j - 1)*M + i - 1] == 128)
525     INPUT[j*M + i] += 1;

        if (DIR[(j - 1)*M + i] == 64)
            INPUT[j*M + i] += 1;

        if (DIR[(j - 1)*M + i + 1] == 32)
            INPUT[j*M + i] += 1;
    }
    int count = 10;
    int count1;
535 while (count > 0)
    {
        count1=0;
        for(j=N-1; j>-1; j--)
        for(i=0; i<M; i++)
        {
            if (INPUT[j*M + i] == 0)
            {
                count1++;
                Q[drainTo[j*M + i]] += Q[j*M + i];
545     INPUT[drainTo[j*M + i]] -= 1;
                INPUT[j*M + i] = 1000;
            }
        }

        count = count1;
    }
    count1=0;
    for(j=N-2; j>0; j--)
    for(i=1; i<M-1; i++)
555 {
        if (INPUT[j*M + i] != 1000)
            count1++;
    }
    for(j=0; j<N; j++)
    for(i=0; i<M; i++)
        Q[j*M + i] *=dx*dy;
}

void functionSet::slopeCalc(double *Z, double *SLOPE)
565 {
    double s[4];
    int i, j;
    for(i = 1; i<M-1; i++)
    for(j = 1; j<N-1; j++)
    {
        s[0] = (Z[j*M + i + 1] - Z[j*M + i - 1]) / (2*dx);
        s[1] = (Z[(j + 1)*M + i] - Z[(j - 1)*M + i]) / (2*dy);
    }
}

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
s[2] = (Z[(j - 1)*M + i + 1] - Z[(j + 1)*M + i - 1]) / (2*sqrt(dx*dx + dy*dy));
s[3] = (Z[(j + 1)*M + i + 1] - Z[(j - 1)*M + i - 1]) / (2*sqrt(dx*dx + dy*dy));
575 SLOPE[j*M + i] = (sqrt(s[0]*s[0] + s[1]*s[1]) + sqrt(s[2]*s[2] + s[3]*s[3])) / 2.0;
}
for(j=0; j<N; j++)
{
    SLOPE[j*M + 0] =SLOPE[j*M + 1];SLOPE[j*M + M - 1] =SLOPE[j*M + M - 2];
}
for(i=0; i<M; i++)
{
    SLOPE[(N-1)*M + i] =SLOPE[(N-2)*M + i];SLOPE[0*M + i] =SLOPE[1*M + i];
}
585 }

void functionSet::curveCalc(double *Z, double *CURVE)
{
    int i, j;
    for(i = 1; i<M-1; i++)
    for(j = 1; j<N-1; j++)
        CURVE[j*M + i] = (Z[j*M + i + 1] + Z[j*M + i - 1] + Z[(j + 1)*M + i] + Z[(j - 1)*M + i] - 4*Z[j*M + i]) / (dx*dx);

    for(j=0; j<N; j++)
595 {
        CURVE[j*M + 0] =CURVE[j*M + 1];CURVE[j*M + M - 1] =CURVE[j*M + M - 2];
    }

    for(i=0; i<M; i++)
    {
        CURVE[(N-1)*M + i] =CURVE[(N-2)*M + i];CURVE[0*M + i] =CURVE[1*M + i];
    }
}
```

Listing C.4 – ‘numeric.cpp’: Matrix Operation and writing data

```
#include "header.h"
int functionSet::temporalBCset(double *X, double *Y, double *Z, string* filename)
{
    int i, j;
    scanTime[0] = 0.25;
    scanTime[1] = 0.50;
7 scanTime[2] = 1.0;
    scanTime[3] = 2.0;
    scanTime[4] = 4.0;
    scanTime[5] = 8.0;
    scanTime[6] = 16.0;
    double temp;
    int mml, nml, ll = 0;
    int nn;
    remove("gradients.ods");
    output2.open("gradients.ods");
17 double aver, avel;
    for (nn = 0; nn < 7; nn++)
    {
        aver = 0;
        avel = 0;
        readLand(X, Y, Z, filename[nn]); //
        replaceError(param, Z);
        // filter(param, param, 1);
        int mml, nml, ll = 0;
        double temp;
27 for(i = 0; i < M ; i++)
    {
        down[nn*M + i] = (param[2*M + i] - param[4*M + i]) / (2*dy);
        up[nn*M + i] = (param[(N - 2)*M + i] - param[(N - 4)*M + i]) / (2*dy);
    }
    for(j = 0; j < N ; j++)
    {
        left[nn*N + j] = (param[j*M + 2] - param[j*M + 4]) / (2*dx) ;
        right[nn*N + j] = (param[j*M + M - 3] - param[j*M + M - 5]) / (2*dx);
        output2 << nn << " " << j << " " << left[nn*N + j] << " " << right[nn*N + j] << endl;
```

```

37 }
   while (ll < 10)
   {
       for (i = 1; i < M-1 ; i++)
       {
           down[nn*M + i] = (down[nn*M + i] + down[nn*M + i + 1] + down[nn*M + i - 1]) / 3.0;
           up[nn*M + i] = (up[nn*M + i] + up[nn*M + i + 1] + up[nn*M + i - 1]) / 3.0;
       }
       for (j = 1; j < N-1 ; j++)
       {
47         right[nn*N + j] = (right[nn*N + j] + right[nn*N + j + 1] + right[nn*N + j - 1]) / 3.0;
           left[nn*N + j] = (left[nn*N + j] + left[nn*N + j + 1] + left[nn*N + j - 1]) / 3.0;
       }
       ll++;
   }
   return 1;
}

int functionSet::dtCalc(double *Q, double *SLOPE)
57 {
   int i, j;
   int k;
   double max = 0, AS;
   double s[4];
   for (i = 1; i < M-1; i++)
   for (j = 1; j < N-1; j++)
   {
       AS = pow(Q[j*M + i], m) * pow(SLOPE[j*M + i], n-1.0) * K[j*M + i] * 1.4142 / dx
           + (pow(abs(D[j*M + i + 1] - D[j*M + i - 1]) / (2*dx) + 1e-5, 0.6666666667)
67       + pow(abs(D[(j + 1)*M + i] - D[(j - 1)*M + i]) / (2*dy) + 1e-5, 0.6666666667)) / pow(dx, 0.6666666667);
       if (AS > max)
           max = AS;
   }
   dt = CFL/max;
   return 1;
}

int functionSet::PHICalc(double *Q, double *SLOPE, double *PHI) //the convection term + noise trem (deposition) + tectonic
{
77 int i, j;

   //////////////////////////////////////////////////attention
   for (i = 0; i < M; i++)
   for (j = 0; j < N; j++)
   {
       PHI[j*M + i] = E - delta / dx * K[j*M + i] * (pow(Q[j*M + i], m) * pow(SLOPE[j*M + i], n) - Omega[j*M + i]);
       //Note: Omega is calculated in the run loop
       if (E - PHI[j*M + i] < 0 || Q[j*M + i] == 0 || SLOPE[j*M + i] == 0 )
           PHI[j*M + i] = E;
87 }
   return 0;
}

int functionSet::RungeKutta(double *Z, double *Zstar, double *Q, double *SLOPE, double *dZstar, double *PHI, double *Y)
{
   int i, j;
   slopeCalc(Z, SLOPE);
   PHICalc(Q, SLOPE, PHI);
   for (i = 0; i < M; i++)
   for (j = 0; j < N; j++)
97 {
       dZstar[j*M + i] = dt * PHI[j*M + i];
       Zstar[j*M + i] = Z[j*M + i] + dZstar[j*M + i];
   }
   slopeCalc(Zstar, SLOPE);
   direction(Zstar, PHI, Y);
   drainageQ(Q);
   PHICalc(Q, SLOPE, PHI);
   for (i = 0; i < M; i++)
   for (j = 0; j < N; j++)
107 Zstar[j*M + i] = Z[j*M + i] + dZstar[j*M + i] / 2.0 + dt * PHI[j*M + i] / 2.0;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
boundarySet(Zstar, Y);
return 0;
}

int functionSet::crankNicholson(double *Zstar, double *Y)
{
    int i, j;
    double f = 1.0;

117 for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
        alpha[j*M + i] = 0.5*dt*D[j*M + i]/(dx*dx);

    for(j = 0; j<N; j++)
    {
        i = 1;
        alpha[j*M + i]*=f;
        i = M - 2;
        alpha[j*M + i]*=f;
127 }

    for(i = 0; i<M; i++)
    {
        j = 1;
        alpha[j*M + i]*=f;
        j = N-2;
        alpha[j*M + i]*=f;
    }

137 for(j = 1; j<N-1; j++)
    {
        i = 1;
        a[j*M + 1] = 0;
        b[j*M + 1] = 1 + alpha[j*M + i]/2.0 + alpha[j*M + i + 1]/2.0; //BC:z = 0 (1 + 2 alpha), BC:dz = 0 (1 + alpha)
        c[j*M + 1] = - alpha[j*M + i]/2 - alpha[j*M + i + 1]/2.0;
        d[j*M + 1] = (alpha[j*M + i]/2.0 + alpha[(j - 1)*M + i]/2.0)*Zstar[(j - 1)*M + i] + (1 - alpha[j*M + i]
            - alpha[(j - 1)*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0)*Zstar[j*M + i]
            + (alpha[j*M + i]/2.0 + alpha[(j + 1)*M + i]/2.0)*Zstar[(j + 1)*M + i] -
            (- alpha[j*M + i]/2 - alpha[j*M + i - 1]/2.0)*leftBC[j] ;
147 i = M - 2;
        a[j*M + M - 2] = - alpha[j*M + i]/2 - alpha[j*M + i - 1]/2.0;
        b[j*M + M - 2] = 1 + alpha[j*M + i]/2.0 + alpha[j*M + i - 1]/2.0; //BC:z = 0 (1 + 2*alpha), BC:dz = 0 (1 + alpha)
        c[j*M + M - 2] = 0;
        d[j*M + M - 2] = (alpha[j*M + i]/2.0 + alpha[(j - 1)*M + i]/2.0)*Zstar[(j - 1)*M + i] +
            (1 - alpha[j*M + i] - alpha[(j - 1)*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0)*Zstar[j*M + i]
            + (alpha[j*M + i]/2.0 + alpha[(j + 1)*M + i]/2.0)*Zstar[(j + 1)*M + i] -
            (- alpha[j*M + i]/2 - alpha[j*M + i + 1]/2.0)*rightBC[j] ;
    }

157 for(j = 1; j<N-1; j++)
    for(i = 2; i<M-2; i++)
    {
        a[j*M + i] = - alpha[j*M + i]/2 - alpha[j*M + i - 1]/2.0;
        b[j*M + i] = 1 + alpha[j*M + i] + alpha[j*M + i - 1]/2.0 + alpha[j*M + i + 1]/2.0 ; //1
        c[j*M + i] = - alpha[j*M + i]/2 - alpha[j*M + i + 1]/2.0;
        d[j*M + i] = (alpha[j*M + i]/2.0 + alpha[(j - 1)*M + i]/2.0)*Zstar[(j - 1)*M + i] +
            (1 - alpha[j*M + i] - alpha[(j - 1)*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0)*Zstar[j*M + i]
            + (alpha[j*M + i]/2.0 + alpha[(j + 1)*M + i]/2.0)*Zstar[(j + 1)*M + i];
    }

167 Thomas(a,b,c,d,Zstar,0);
boundarySet(Zstar, Y);
for(i = 1; i<M-1; i++)
{
    j = 1;
    a[1*M + i] = 0;
    b[1*M + i] = 1 + alpha[j*M + i]/2.0 + alpha[(j + 1)*M + i]/2.0 ; //BC:z = 0 (1 + 2*alpha), BC:dz = c (1 + alpha)
    //the value of c is epsilon or alpha*(mainSlope*dy (it is mentioned in d(i,j)
    c[1*M + i] = - alpha[j*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0;
177 d[1*M + i] = (alpha[j*M + i]/2.0 + alpha[j*M + i - 1]/2.0)*Zstar[j*M + i-1] +
    (1 - alpha[j*M + i] - alpha[j*M + i - 1]/2.0 - alpha[j*M + i + 1]/2.0)*Zstar[j*M + i] +
```

```

        (alpha[j*M + i]/2.0 + alpha[j*M + i + 1]/2.0)*Zstar[j*M + i+1] -
        (- alpha[j*M + i]/2.0 - alpha[(j - 1)*M + i]/2.0)*downBC[i];

j = N-2;
a[(N-2)*M + i] = - alpha[j*M + i]/2.0 - alpha[(j - 1)*M + i]/2.0;
b[(N-2)*M + i] = 1 + alpha[j*M + i]/2.0 + alpha[(j - 1)*M + i]/2.0 ; //BC:z = 0 (1 + 2*alpha), BC:dz = 0 (1 + alpha)
c[(N-2)*M + i] = 0;
d[(N-2)*M + i] = (alpha[j*M + i]/2.0 + alpha[j*M + i - 1]/2.0)*Zstar[j*M + i-1] +
    (1 - alpha[j*M + i] - alpha[j*M + i - 1]/2.0 - alpha[j*M + i + 1]/2.0)*Zstar[j*M + i] +
    (alpha[j*M + i]/2.0 + alpha[j*M + i + 1]/2.0)*Zstar[j*M + i+1] -
    (- alpha[j*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0)*upBC[i];
}
i = M/2; j = 1;
{
    a[j*M + i] = 0;
    b[j*M + i] = 1 + alpha[j*M + i] + alpha[(j - 1)*M + i]/2.0 + alpha[(j + 1)*M + i]/2.0;
    //BC:z = 0 (1 + 2*alpha), BC:dz = 0 (1 + alpha)
    c[j*M + i] = - alpha[j*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0;
    d[j*M + i] = (alpha[j*M + i]/2.0 + alpha[j*M + i - 1]/2.0)*Zstar[j*M + i-1] + (1 - alpha[j*M + i] -
197     alpha[j*M + i - 1]/2.0 - alpha[j*M + i + 1]/2.0)*Zstar[j*M + i] +
        (alpha[j*M + i]/2.0 + alpha[j*M + i + 1]/2.0)*Zstar[j*M + i+1];
}
for(i = 1; i<M-1; i++)
for(j = 2; j<N-2; j++)
{
    a[j*M + i] = - alpha[j*M + i]/2.0 - alpha[(j - 1)*M + i]/2.0;
    b[j*M + i] = 1 + alpha[j*M + i] + alpha[(j - 1)*M + i]/2.0 + alpha[(j + 1)*M + i]/2.0;
    c[j*M + i] = - alpha[j*M + i]/2.0 - alpha[(j + 1)*M + i]/2.0;
    d[j*M + i] = (alpha[j*M + i]/2.0 + alpha[j*M + i - 1]/2.0)*Zstar[j*M + i-1] +
207     (1 - alpha[j*M + i] - alpha[j*M + i - 1]/2.0 - alpha[j*M + i + 1]/2.0)*Zstar[j*M + i] +
        (alpha[j*M + i]/2.0 + alpha[j*M + i + 1]/2.0)*Zstar[j*M + i+1];
}
Thomas(a,b,c,d,Zstar,1);
boundarySet(Zstar, Y);
return 0;
}

int functionSet::Thomas(double *a,double *b,double *c,double *d,double *Z, int rowCol)
217 {
    int i,j;
    if(rowCol==0)
    {
        for(j = 1; j<N-1; j++)
        {
            c[j*M + 1] = c[j*M + 1]/b[j*M + 1];
            d[j*M + 1] = d[j*M + 1]/b[j*M + 1];
            b[j*M + 1] = 1.0;
            for(i = 2; i <= M-2; i++)
227 {
                c[j*M + i] = c[j*M + i]/(b[j*M + i] - a[j*M + i]*c[j*M + i - 1]);
                d[j*M + i] = (d[j*M + i] - a[j*M + i]*d[j*M + i - 1])/(b[j*M + i] - a[j*M + i]*c[j*M + i - 1]);
                b[j*M + i] = 1.0;
            }
            Z[j*M + M - 2] = d[j*M + M - 2];
            for(i = M - 3; i >= 1; i--)
                Z[j*M + i] = d[j*M + i] - c[j*M + i]*Z[j*M + i + 1];
        }
    }
237 if(rowCol==1)
    {
        for(i = 1; i<M-1; i++)
        {
            c[1*M + i] = c[1*M + i]/b[1*M + i];
            d[1*M + i] = d[1*M + i]/b[1*M + i];
            b[1*M + i] = 1.0;
            for(j = 2; j <= N-2; j++)
            {
                c[j*M + i] = c[j*M + i]/(b[j*M + i] - a[j*M + i]*c[(j - 1)*M + i]);
                d[j*M + i] = (d[j*M + i] - a[j*M + i]*d[(j - 1)*M + i])/(b[j*M + i] - a[j*M + i]*c[(j - 1)*M + i]);
                b[j*M + i] = 1.0;
            }
        }
    }
247 }
}

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
Z[(N - 2)*M + i] = d[(N - 2)*M + i];
for(j = N - 3; j >= 1; j--)
    Z[j*M + i] = d[j*M + i] - c[j*M + i]*Z[(j + 1)*M + i];
}
}
return 0;
}

257 int functionSet::addNoise(double *Z, double dt)
{
    double random, rand1;
    double noise;
    int i, j;
    std::random_device rd;
    std::default_random_engine generator(rd());
    std::cauchy_distribution<double> cauchy(0, 0.11757900431008157);
    for(i = 1; i < M-1; i++) //adding the noise into the field
267 for(j = 1; j < N-1; j++)
    {
        random = 10;
        while (random > 1)
            random = abs(cauchy(generator));
        noise = noiseAve*random*dt;
        rand1 = (double) rand() / (RAND_MAX);
        if(rand1 > 0.5)
            Z[j*M + i] = Z[j*M + i] + noise;
        else if(rand1 < 0.5)
277 Z[j*M + i] = Z[j*M + i] - noise;
    }
    return 0;
}

int functionSet::boundarySet(double *Z, double *Y)
{
    int i, j;
    double timeGap = scanTime[BCstep + 1] - scanTime[BCstep];
    double temp;
287 for(i = 1; i < M - 1; i++)
    {
        temp = down[BCstep*M + i] + (down[(BCstep + 1)*M + i] - down[BCstep*M + i])/timeGap*(t - scanTime[BCstep]);
        if(temp > 0 )
            downBC[i] = temp*dy; //attention downBC = 0
        else
            downBC[i] = 0;
        temp = Z[1*M + i] + downBC[i];
        if(temp <= Z[0*M + i])
            Z[0*M + i] = temp;
297 else if (Z[0*M + i] > Z[1*M + i])
            downBC[i] = Z[0*M + i] - Z[1*M + i];
        else
            downBC[i] = 0;
        temp = up[BCstep*M + i] + (up[(BCstep + 1)*M + i] - up[BCstep*M + i])/timeGap*(t - scanTime[BCstep]);
        if(temp > 0 )
            upBC[i] = temp*dy;
        else
            upBC[i] = 0;
        temp = Z[(N - 2)*M + i] + upBC[i];
307 if(temp <= Z[(N - 1)*M + i])
            Z[(N - 1)*M + i] = temp;
        else if (Z[(N - 1)*M + i] > Z[(N - 2)*M + i])
            upBC[i] = Z[(N - 1)*M + i] - Z[(N - 2)*M + i];
        else
            upBC[i] = 0;
    }

    for(i = M/2-int(200/dx); i < M/2+int(200/dx); i++)
317 {
        downBC[i] = 0.0;
        Z[0*M + i] = Z[1*M + i] + downBC[i];
    }
}
```

```

for(j = 1; j < N - 1 ; j++)
{
temp = left[BCstep*N + j] + (left[(BCstep + 1)*N + j] - left[BCstep*N + j])/timeGap*(t - scanTime[BCstep]);
if(temp > 0 )
leftBC[j] = temp*dx;
else
327 leftBC[j] = 0;
temp = Z[j*M + 1] + leftBC[j];

if(temp <= Z[j*M + 0])
Z[j*M + 0] = temp;
else if (Z[j*M + 0] > Z[j*M + 1])
leftBC[i] = Z[j*M + 0] - Z[j*M + 1];
else
leftBC[i] = 0;
temp = right[BCstep*N + j] + (right[(BCstep + 1)*N + j] - right[BCstep*N + j])/timeGap*(t - scanTime[BCstep]);
337 if(temp > 0 )
rightBC[j] = temp*dx;
else
rightBC[j] = 0;

temp = Z[j*M + M - 2] + rightBC[j];

if(temp <= Z[j*M + M - 1])
Z[j*M + M - 1] = temp;
else if (Z[j*M + M - 1] > Z[j*M + M - 2])
347 leftBC[i] = Z[j*M + M - 1] - Z[j*M + M - 2];
else
leftBC[i] = 0;
}

Z[0*M + 0] = (Z[1*M + 0] + Z[0*M + 1])/2;
Z[0*M + M - 1] = (Z[1*M + M - 1] + Z[0*M + M - 2])/2;
Z[(N - 1)*M + 0] = (Z[(N - 1)*M + 1] + Z[(N - 2)*M + 0])/2;
Z[(N - 1)*M + M - 1] = (Z[(N - 2)*M + M - 1] + Z[(N - 1)*M + M - 2])/2;

for(i = 0; i < M; i++)
357 for(j = 0; j < N; j++)
if (Z[j*M + i] < mainSlope*Y[j*M + i]-25/cos(atan(mainSlope)))
Z[j*M + i] = mainSlope*Y[j*M + i]-25/cos(atan(mainSlope));
Z[0*M + M/2] = 0;
return 0;
}

int functionSet::replaceError(double *Z, double *Zstar)//Z* to Z
{
int i, j;
367 error = 0;
for(i = 0; i < M; i++)
for(j = 0; j < N; j++)
{
Z[j*M + i] = Zstar[j*M + i];
}
return 0;
}

int functionSet::writeData(double *X, double *Y, double *Z, double *X1, double *X2, double *X3, double *X4,
377 double *X5, double *X6, double *X7, string folder, int step)
{
int i, j;
char ch[5];
double L = 3;
d2s(double(step), ch, L);
string str1, str2, str3;
str1 = folder;
str1 += "/DATA";
str1 += ch;
387 str1 += ".csv";
remove(str1.c_str());
output1.open(str1.c_str(), ios::app);
double energy = 0;
cout << "-----WRITING TO: " << folder << "-----" << endl;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
output1<<" X,      Y,      Z,      H, slope,Q, AREA, L,  curve,power, downL" <<endl;

for(i = 0; i<M; i++)
for(j = 0; j<N; j++)
{
397 output1 << X[j*M + i] << " ,      " << Y[j*M + i] << " ,      " << Z[j*M + i] << " ,      " <<X1[j*M + i] << " ,      "
    <<X2[j*M + i]<<" ,      " <<X3[j*M + i]<<" ,      " <<X4[j*M + i] <<" ,      " <<X5[j*M + i] <<" ,      " <<X6[j*M + i]
    <<" ,      " <<X2[j*M + i]*X3[j*M + i] <<" ,      " <<X7[j*M + i] << endl;
}
for(i = 0; i<M; i++)
for(j = 0; j<N; j++)
{
    if(DIR[j*M + i] == 1 || DIR[j*M + i] == 4 || DIR[j*M + i] == 16 || DIR[j*M + i] == 32)
    {
        energy += sqrt(X3[j*M + i])*dx;
407 }
    else
    {
        energy += sqrt(X3[j*M + i]*2.0)*dx;
    }
}
cout << "energy (Q) = " <<energy<<endl;
energy = 0;
for(i = 0; i<M; i++)
for(j = 0; j<N; j++)
417 {
    if(DIR[j*M + i] == 1 || DIR[j*M + i] == 4 || DIR[j*M + i] == 16 || DIR[j*M + i] == 32)
    {
        energy += sqrt(X4[j*M + i])*dx;
    }
    else
    {
        energy += sqrt(X4[j*M + i]*2.0)*dx;
    }
}
427 cout << "energy (A) = " <<energy<<endl;
output1.close();
return 0;
}

int functionSet::writeXYZ(double *X, double *Y, double *Z, string folder, int step)
{
    int i,j;
    char ch[5];
    double L = 3;
437 d2s(double(step),ch,L);
    string str1, str2, str3;
    str1 = folder;
    str1 += "/XYZ";
    str1 += ch;
    str1 += ".dat";
    remove(str1.c_str());
    output1.open(str1.c_str(),ios::app);
    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
447 {
        output1 << X[j*M + i] << "      " << Y[j*M + i] << "      " << Z[j*M + i] <<endl;
    }
    output1.close();
}

int functionSet::analytic(double *X, double *Y, double *Z)
{
    double numb = 200;
    int k, i,j;
457 remove("analytics.csv");
    output1.open("analytics.csv",ios::app);
    output1<<" X,      Y,      Z,      H" <<endl;
    for(i = 0; i<M; i++)
    for(j = 0; j<N; j++)
    {
```

```

Z[j*M + i] = (1-X[j*M + i]*X[j*M + i]) / 2.0;
for(k = 1; k<numb; k = k+2)
    Z[j*M + i] += -16/(PI*PI*PI)*( sin(k*PI*(1+X[j*M + i]) / 2) / (k*k*k*sinh(k*PI)) *(sinh(k*PI*(1+Y[j*M + i]) / 2) +
sinh(k*PI*(1-Y[j*M + i]) / 2)));
467 output1 << X[j*M + i] << " , " << Y[j*M + i] << " , " << Z[j*M + i] << " , " << Z[j*M + i] << " " << endl;
}
output1.close();
return 0;
}

void functionSet::d2s(double num, char s[], unsigned int width)
{
    int i, j;
    //---width; // allow room for \0.
477 double decimal = abs(num) - abs(int(num));
    int integer = abs(int(num));
    deque<char> result;
    if (integer == 0)
    {result.push_front('0');}
    while (integer != 0)
    {
        result.push_front(integer % 10 + '0');
        integer /= 10;
    }
487 if (num < 0) // check for negative value
    result.push_front('-');
    result.push_back('.');

    while (decimal != 0.00 && result.size() < width)
    {
        decimal *= 10.0;

        result.push_back(int(decimal) % 10 + '0');
        decimal = decimal - int(decimal);
497 }

    while (result.size() < width)
        result.push_front('0'); // pad with spaces.
    for (int i=0; i<width; ++i) // results into array s
        s[i] = result[i];
    s[width] = '\0';
}

```

Listing C.5 – ‘RUN.cpp’: Numerical modeling

```

#include "header.h"

3 int functionSet::RUN(double *X, double *Y, double *Z, double *Zstar, double *Q, double *Qx, double *Qy,
    double *SLOPE, double *CURVE, double *PHI, double *dZstar, double *AREA, double *LL, double *downL,
    string addressRUN, double initialTime, double finalTime, string initialTimeZ, string finalTimeZ)
{
    double *order;
    double random, rand1;
    double noise;
    default_random_engine generator;
    normal_distribution<double> distribution(0,0.503702461771);//The noise distribution
    std::cauchy_distribution<double> cauchy(0,0.11757900431008157);
13 int i, j;
    order = (double*) calloc(M*N , sizeof(double));
    //check the writedata function, it has been changed probably
    string makefolder = "mkdir " + addressRUN;
    system(makefolder.c_str());

    //now it is constant rainfall, when you change the setting delete this sentence
    double epsilon0 = 0.1;
    epsilon = epsilon0;
    t = initialTime;
23 double step = (finalTime - initialTime) / 10.0;
    double captureTime = t * 2.0;
    int nn;
    for (nn = 0; nn < 7; nn++)

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
    if( initialTime == scanTime[nn])
        BCstep = nn;

    double BCtime = scanTime[BCstep + 1] ;
    defineD(varD);
    defineK(0);
33 defineOmega(1);
    readLand(X, Y, Z, initialTimeZ);///
    //filter33(Z, Z, 1);
    //writeDATA(X, Y, Z, R, D, addressRUN, 1000);
    replaceError(Zstar, Z);
    boundarySet(Zstar, Y);
    boundarySet(Z, Y);
    direction(Zstar, PHI, Y);
    drainageQ(Q);
    slopeCalc(Z, SLOPE);
43 //we use the original data to calculate slope (without filling)
    double ERROR = 5;
    int num = 0;
    //drainageAREAandLL(AREA, LL);
    curveCalc(Z, CURVE);
    drainageAREAandLL(AREA, LL, order);
    writeDATA(X, Y, Z, Z, SLOPE, Q, AREA, LL, CURVE, downL, addressRUN, num);
    writeXYZ(X, Y, Z, addressRUN, num);

    while(t < finalTime )
53 {
        boundarySet(Z, Y);

        dtCalc(Q, SLOPE);
        epsilon = epsilon0 - (epsilon0 - epsilon0/10.0)/(finalTime - initialTime)*t; //pow(10, rndCoef)/2.0;
        //epsilon0 - (epsilon0 - epsilon0/10.0)/(finalTime - initialTime)*t; //is used in filling processes
        if(t + dt > captureTime)
            dt = captureTime - t;
        if(t + dt > BCtime)
        {
63 BCstep++; BCtime = scanTime[BCstep + 1];
        }
        RungeKutta(Z, Zstar, Q, SLOPE, dZstar, PHI, Y);
        crankNicholson(Zstar, Y);
        replaceError(Z, Zstar);
        //addNoise(Z, dt);
        defineOmega(1);
        boundarySet(Z, Y);
        boundarySet(Zstar, Y);
        t+=dt;
73 //ERROR = error;
        //cout <<"CHANGE (MM) =" << ERROR << " " << "t =" << t << endl;
        cout << "t =" << t << endl;
        if(t == captureTime)
        {
            direction(Zstar, PHI, Y); //these two function are also in numeric PHICalc
            drainageQ(Q);
            slopeCalc(Z, SLOPE);
            curveCalc(Z, CURVE); Qcomponents(X, Y, Q, Qx, Qy);
            drainageAREAandLL(AREA, LL, order);
83 writeDATA(X, Y, Z, Z, SLOPE, Q, AREA, LL, CURVE, downL, addressRUN, ++num);
            writeXYZ(X, Y, Z, addressRUN, num);
            captureTime*=2.0;
        }

        direction(Zstar, PHI, Y); //these two function are also in numeric PHICalc
        drainageQ(Q);
        slopeCalc(Z, SLOPE);
    }
    readLand(X, Y, Zstar, finalTimeZ);
93 ERROR = 0;
    cout << "I AM HERE" << endl;
    ERROR = correlation(Zstar, Z, M*N);
    cout << "—————ERROR = " << ERROR << endl << endl ;
    std::string command = "python PlotScript.py";
```

```

        system(command.c_str());
    return 1;
}

```

Listing C.6 – ‘optimization.cpp’: Different objective functions and optimisation

```

#include "header.h"
double functionSet::stepRun(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE,
    double *PHI, double *dZstar, double initialT, double finalT, string finalZ) //Z is the initial Z
{
    epsilon = 1.0;
    double random,rand1;
    double noise;
8   int i,j;
    defineRandom_Noise_D_K_Omega(Z,0,0,0,0);
    replaceError(Zstar, Z);
    direction(Zstar, PHI, Y);
    drainageQ(Q);
    slopeCalc(Z,SLOPE); //we use the original data to calculate slope (without filling)
    t = initialT;
    int nn;
    for (nn = 0; nn < 7; nn++)
        if ( initialT == scanTime[nn])
18     BCstep = nn;
    double BCtime = scanTime[BCstep + 1] ;
    while(t < finalT)
    {
        dtCalc(Q, SLOPE);
        epsilon = 0.05 - (0.05 - 1e-2)/finalT*t; //rndCoef*pow(10,0.5)*dt
        if (dt < 0.5e-3)
            return 1/dt;
        if (dt + t > finalT)
            dt = finalT - t;
28     if(t + dt > BCtime)
        {
            BCstep++; BCtime = scanTime[BCstep + 1];
        }
        RungeKutta(Z, Zstar, Q, SLOPE, dZstar, PHI, Y);
        crankNicholson(Zstar, Y);
        replaceError(Z, Zstar);
        //defineOmega(1);
        //addNoise(Z,dt);
38     defineRandom_Noise_D_K_Omega(Z,1,0,0,0);
        boundarySet(Z, Y);
        boundarySet(Zstar, Y);
        t+=dt;
        //cout << " " << "t =" << t << endl;
        if (Kave == 0.0)
            continue;
        direction(Zstar, PHI, Y);
        drainageQ(Q);
        slopeCalc(Z,SLOPE);
48     }

    readLand(X, Y, Zstar,finalZ);
    return meanSquareError(Z, Zstar, M*N);
}

objectives functionSet::multiObjective_Zerror_discharge(double *X, double *Y, double *Z, double *Zstar, double *Q,
    double *SLOPE, double *PHI, double *dZstar, string initialTime, string finalTime, string* filename)
{
    int i,j,k;
58     double *Qs,*Ls;
    double rmsZ, rmsQ, rmsL;
    objectives ERROR;
    ERROR.a = 0;
    ERROR.b = 0;
    ERROR.c = 0;
    ERROR.d = 0;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
readLand(X, Y, Z, initialTime);
//MCH.smoothZ(Z);
//cout<<"ok"<<endl;
68 double time[] = {0.25,0.5,1.0,2.0,4.0,8.0};
Qs = new double[6*M*N];
Ls = new double[6*M*N];
for (k = 1; k<6; k++)
{
    rmsZ = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, time[k-1], time[k], filename[k]); //0:uniform D
    drainageAREAandLL(SLOPE, PHI, dZstar);
    //because we don't need SLOPE and PHI at this step, they are used to save AREA and LL. AREA:SLOPE, LL:PHI, order:dZstar
    for (j = 0; j<N; j++)
    for (i = 0; i<M; i++)
78 {
        Qs[k*M*N+j*M + i] = Q[j*M + i];
        Ls[k*M*N+j*M + i] = PHI[j*M + i];
    }
}
ERROR.a = rmsZ;
output1.close();
output1.open("Q_model.csv", ios::trunc);

for (j = 0; j<N; j++)
88 for (i = 0; i<M; i++)
{
    for (k = 1; k<6; k++)
        output1 << Qs[k*M*N+j*M + i] << ', ';
    output1 << endl;
}

//calculating Q rms
output1.close();
//Running the python script to calculate rms of discharges
98 sleep_for(seconds(4));
std::string command = "python Q_RMS.py";
system(command.c_str());
sleep_for(seconds(4));
input1.open("Q_RMS.dat"); //
input1>>rmsQ;
ERROR.b = rmsQ;
input1.close();
remove("Q_RMS.dat");
remove("Q_model.csv");
108 //calculating L rms
output1.close();
output1.open("L_model.csv", ios::trunc);
for (j = 0; j<N; j++)
for (i = 0; i<M; i++)
{
    for (k = 1; k<6; k++)
        output1 << Ls[k*M*N+j*M + i] << ', ';
    output1 << endl;
}
118 output1.close();
//Running the python script to calculate rms of discharges
sleep_for(seconds(4));
command = "python L_RMS.py";
system(command.c_str());
sleep_for(seconds(4));
input1.open("L_RMS.dat"); //
input1>>rmsL;
ERROR.c = rmsL;
input1.close();
128 remove("L_RMS.dat");
remove("L_model.csv");

ERROR.d = 0; //rmsL*rmsQ*rmsZ;

return ERROR;
}
```

```

//Objective function is based on the exceedance probability of discharge at 8h
double functionSet::objective_discharge(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE,
138 double *PHI, double *dZstar, string initialTime, string finalTime, string* filename)
{
    int i, j;
    readLand(X, Y, Z, initialTime);
    //MCH.smoothZ(Z);
    double ERROR = 0;
    double error1;
    //error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 0.25, 8.0, filename[5]); //0:uniform D

    //Writing the model Q data in the file
148 output1.close();
    output1.open("model_discharge.csv", ios::trunc);
    for(j = 0; j < N; j++)
    for(i = 0; i < M; i++)
    {
        output1 << Q[j*M + i] << endl;
    }
    output1.close();
    //Running the python script to calculate rms of discharges
    sleep_for(seconds(4));
158 std::string command = "python Q-RMS.py";
    system(command.c_str());
    sleep_for(seconds(4));
    input1.open("Discharge_RMS.dat"); //
    input1 >> ERROR;
    input1.close();
    remove("Discharge_RMS.dat");
    remove("model_discharge.csv");
    return ERROR;
}

168 double functionSet::objective1(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI,
    double *dZstar, string initialTime, string finalTime, string* filename)
{
    readLand(X, Y, Z, initialTime);
    //MCH.smoothZ(Z);
    double ERROR = 0;
    double error1 = 0;
    error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 0.25, 0.5, filename[1]); //0:uniform D
    output1 << " " << error1;
178 //ERROR += error1;
    error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 0.5, 1.0, filename[2]);
    output1 << " " << error1;
    //ERROR += error1;
    error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 1.0, 2.0, filename[3]);
    output1 << " " << error1;
    //ERROR += error1;
    error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 2.0, 4.0, filename[4]);
    output1 << " " << error1;
    //ERROR += error1;
188 error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 4.0, 8.0, filename[5]);
    output1 << " " << error1;
    ERROR += error1;
    //error1 = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 8.0, 16.0, filename[6]);
    //output1 << " " << error1;
    //ERROR += error1;
    //ERROR += stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, 0.25, 16.0,
    // "/home/mohsen/LandScapeCode/input/exp8/16h00min_8mm.dat"); //number "1" is for var D and "0" for fixed D
    return ERROR;
}

198 double functionSet::objective2(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE,
    double *PHI, double *dZstar)
{
    int i, j;
    epsilon = 0.05;
    definedD(varD);
    readLand(X, Y, Z, "/home/mohsen/LandScapeCode/input/exp8/16h00min_8mm.dat");
    //MCH.smoothZ(Z);

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
boundarySet(Z, Y);
208 //It will be changed during the filling process, it is just the initial value
replaceError(Zstar, Z);
direction(Zstar, PHI, Y);
drainageQ(Q);
slopeCalc(Zstar, SLOPE);
error = 0;
double error1 = 0;
for(i = 1; i<M-1; i++)
for(j = 1; j<N-1; j++)
{
218 error1 = 0;
error1 += D[j*M + i]*(Z[j*M + i + 1] + Z[j*M + i - 1] + Z[(j + 1)*M + i] + Z[(j - 1)*M + i] -
4*Z[j*M + i])/(dx*dx) + (D[j*M + i+1] - D[j*M + i])*(Z[j*M + i + 1] - Z[j*M + i])/(dx*dx)
+ (D[(j + 1)*M + i] - D[j*M + i])*(Z[(j + 1)*M + i] - Z[j*M + i])/(dy*dy);

if(pow(Q[j*M + i], m)*pow(SLOPE[j*M + i], n) > OmegaAve)
error1 += E - delta/dx*K[j*M + i]*(pow(SLOPE[j*M + i], n)*pow(Q[j*M + i], m) - Omega[j*M + i]);
error += error1*error1;
}
return sqrt(error/(M*N));
228 }

int functionSet::calibratePSO(double *X, double *Y, double *Z, double *Zstar, double *Q, double *SLOPE, double *PHI,
double *dZstar, string addressOPT, string initialTime, string finalTime, string* filename)
{
int i, j;
remove(addressOPT.c_str());
output1.open(addressOPT.c_str(), ios::app);
objectives objFun;
int nvar = 7;
double min[nvar];
238 min[0] = 16000; //D,
min[1] = 0.17; //K,
min[2] = 10; //Tc
min[3] = 0; //gamma,
min[4] = 0.5; //m,
min[5] = 1.0; //n,
min[6] = 0.0; //n,
double minOLD[nvar];
double maxOLD[nvar];
double max[nvar];
248 max[0] = 21000; //D,
max[1] = 0.23; //K,
max[2] = 10; //Tc
max[3] = 0; //gamma,
max[4] = 0.5; //m,
max[5] = 1.0; //n,
max[6] = 100.0; //randCoef,
E = 0; //E
int pop = 50;
double *Point, *obj, *selfOptPlace, *vel, *selfOpt;
258 Point = (double*)calloc(pop*nvar, sizeof(double));
selfOptPlace = (double*)calloc(pop*nvar, sizeof(double));
vel = (double*)calloc(pop*nvar, sizeof(double));
obj = (double*)calloc(pop, sizeof(double));
selfOpt = (double*)calloc(pop, sizeof(double));
double optGeneral = 1e6;
double optGeneralPlace[] = {1000, 1000, 0, 0, 0.5, 1.0, 100};
int nn, nn;
double random;
for(nn = 0; nn<pop; nn++)
268 {
//random = (double)rand()/(double)RAND_MAX;
for(nn = 0; nn < nvar; nn++)
{
random = (double)rand()/(double)RAND_MAX;
Point[nn*nvar + nn] = min[nn] + random*(max[nn] - min[nn]); //D
vel[nn*nvar + nn] = 0;
selfOptPlace[nn*nvar + nn] = Point[nn*nvar + nn];
selfOpt[nn] = 1e15;
}
}
```

```

278 vel [mm*nvar + 1] = 0;
    selfOptPlace [mm*nvar + 1] = Point [mm*nvar + 1];
}
for (nn = 0; nn < nvar; nn++)
    output1 << min [nn] << " " << max [nn] << endl;
int iter = 0;
while ( iter < 100)
{
    for (mm = 0; mm < pop; mm++)
    {
288 cout << iter << " " << mm;
        for (nn = 0; nn < nvar; nn++)
            cout << " " << Point [mm*nvar + nn];
        output1 << mm << " ";
        for (nn = 0; nn < nvar; nn++)
            output1 << " " << Point [mm*nvar + nn];
        Dave = Point [mm*nvar + 0];
        Kave = Point [mm*nvar + 1];
        OmegaAve = Point [mm*nvar + 2];
        gamma = Point [mm*nvar + 3];
298 m = Point [mm*nvar + 4];
        n = Point [mm*nvar + 5];
        noiseAve = Point [mm*nvar + 6];
        E = 0;
        //obj [mm] = objective_discharge(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, initialTime, finalTime, filename );
        //obj [mm] = objective2(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar);
        objFun = multiObjective_Zerror_discharge(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar,
        initialTime, finalTime, filename ); obj [mm] = objFun.a*objFun.b;
        if (obj [mm] < optGeneral)
        {
308 optGeneral = obj [mm];
            for (nn = 0; nn < nvar; nn++)
                optGeneralPlace [nn] = Point [mm*nvar + nn];
        }
        if (obj [mm] < selfOpt [mm])
        {
            selfOpt [mm] = obj [mm];
            for (nn = 0; nn < nvar; nn++)
                selfOptPlace [mm*nvar + nn] = Point [mm*nvar + nn];
        }
318 cout << " " << objFun.a << " " << objFun.b << " " << obj [mm] << endl;
        output1 << " " << objFun.a << " " << objFun.b << " " << obj [mm] << endl;
    }
    for (nn = 0; nn < nvar; nn++)
    {
        for (mm = 0; mm < pop; mm++)
        {
            vel [mm*nvar + nn] = vel [mm*nvar + nn] +
            2*(double)rand()/(double)RAND_MAX*1*(selfOptPlace [mm*nvar + nn] - Point [mm*nvar + nn]) +
            2*(double)rand()/(double)RAND_MAX*1*(optGeneralPlace [nn] - Point [mm*nvar + nn]) ;
328 Point [mm*nvar + nn] += vel [mm*nvar + nn];
            if (Point [mm*nvar + nn] > max [nn])
            {
                Point [mm*nvar + nn] = min [nn] + (double)rand()/(double)RAND_MAX*(max [nn] - min [nn]) ;
                //min [nn] + (double)rand()/(double)RAND_MAX*(max [nn] - min [nn]);
                //DO NOT use the limits point = max or min!!!!
                //vel [mm*5 + nn] = 0;
            }
            if (Point [mm*nvar + nn] < min [nn])
            {
338 Point [mm*nvar + nn] = min [nn] + (double)rand()/(double)RAND_MAX*(max [nn] - min [nn]) ;
                //min [nn] + (double)rand()/(double)RAND_MAX*(max [nn] - min [nn]);
                //DO NOT use the limits point = max or min!!!!
                //vel [mm*5 + nn] = 0;
            }
        }
    }
    cout << endl << endl << endl << iter << " " << optGeneral << endl;
    output1 << iter << " " << optGeneral << endl;
    iter++;
348 for (nn = 0; nn < nvar; nn++)

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
{
    output1 << optGeneralPlace[nm] << endl;
    cout << optGeneralPlace[nm] << endl;
}
return 1;
}

void functionSet::SetCursorPos(int XPos, int YPos)
{
358 printf("\033[%d;%dH", YPos+1, XPos+1);
}

int functionSet::confidenceInterval(double q[], double percentage[], double *X, double *Y, double *Z,
double *Zstar, double *Q, double *SLOPE, double *PHI, double *dZstar, double T1, double T2,
string initialTime, string finalTime) //This function uses the value of z in the model to
calculate the confidence interval using the method described in the uncertainty book
{
    int i, j;
    double *Z1, *Z2, *sens, *sensT;
368 double a, sigma;
    Z1 = new double[M*N];
    Z2 = new double[M*N];
    sens = new double[M*N*3];
    sensT = new double[M*N*3];
    Dave = q[0];
    Kave = q[1];
    OmegaAve = q[2];
    m = 0.5;
    n = 1.0;
378 readLand(X, Y, Z, initialTime);
    a = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, T1, T2, finalTime);
    replaceError(Z1, Z);
    readLand(X, Y, Z, finalTime);
    sigma = meanSquareError(Z1, Z, M*N/2)*sqrt(M*N/2)/sqrt(M*N/2 - 3);
    int nm, mm;
    cout << "D = " << q[0]<<" "<<" sigma = " << sigma << " t = 1.9695 (95%)" <<endl;
    cout << "K = " << q[1]<<" "<<" sigma = " << sigma << " t = 1.9695 (95%)" <<endl;
    cout << "Tc = " << q[2]<<" "<<" sigma = " << sigma << " t = 1.9695 (95%)" <<endl;
    cout << "dq1 %" << " dq2 %" << " dq3 %" << " sigmaD" << " sigmaK " << " sigmaTc " << endl;
388
    double range = 0.05;
    double dq ;
    for(i = 0; i < 125; i++)
    {
        dq = -range+(double)rand()/(double)RAND_MAX*2*range;
        cout << dq*100;
        Dave = q[0]*(1 + dq);
        Kave = q[1];
        OmegaAve = q[2];
398 readLand(X, Y, Z, initialTime);
        a = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, T1, T2, finalTime);
        replaceError(Z2, Z);

        for (mm = 0; mm < M*N/2; mm++)
        {
            sens[0*M*N + mm] = (Z2[mm] - Z1[mm]) / (q[0]*dq);
        }
        dq = -range+(double)rand()/(double)RAND_MAX*2*range;
        cout << " " << dq*100;
408 Dave = q[0];
        Kave = q[1]*(1 + dq);
        OmegaAve = q[2];
        readLand(X, Y, Z, initialTime);
        a = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, T1, T2, finalTime);
        replaceError(Z2, Z);
        for (mm = 0; mm < M*N/2; mm++)
        {
            sens[1*M*N + mm] = (Z2[mm] - Z1[mm]) / (q[1]*dq);
        }
418 dq = -range+(double)rand()/(double)RAND_MAX*2*range;
        cout << " " << dq*100;
```

```

Dave = q[0];
Kave = q[1];
OmegaAve = q[2]*(1 +dq);
readLand(X, Y, Z, initialTime);
a = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, T1, T2, finalTime);
replaceError(Z2, Z);
for (nm = 0; nm < M*N/2; nm++)
{
428   sens[2*M*N + nm] = (Z2[nm] - Z1[nm]) / (q[2]*dq);
}
transpose(sens, sensT, M*N/2, 3);
multiple(sensT, sens, Z, 3, M*N/2, M*N/2, 3);
//for (int i = 0; i<3;i++)
// cout << "matrix:"<< Z[0*3 + i] << " "<<Z[1*3 + i] << " "<<Z[2*3 + i] <<endl;
matrixInversion(Z, 3, SLOPE);
cout << " "<<1.9695*sigma*sqrt(SLOPE[0*3 + 0]) << " "<<1.9695*sigma*sqrt(SLOPE[1*3 + 1])
<< " "<< 1.9695*sigma*sqrt(SLOPE[2*3 + 2]) << endl;
}
438 output1.close();
delete [] Z1;
delete [] Z2;
delete [] sens;
delete [] sensT;
return (0);
}

int functionSet::confidenceInterval1(double q0[], double percentage[], double *X, double *Y, double *Z, double *Zstar,
double *Q, double *SLOPE, double *PHI, double *dZstar, double T1, double T2, string initialTime, string finalTime)
448 //This function uses the objective function and second order term to calculate the confidence interval
//(the method described by Prof.Davison)
{
int Samp_Pop = 200;
int i, j;
double *Z1, *Z2, *sens, *sensT, *q,*sigma;
double a, sigma0, sigmaBAR;
Z1 = new double[M*N];
Z2 = new double[M*N];
q = new double[3*Samp_Pop];
458 sigma = new double[Samp_Pop];
sens = new double[Samp_Pop*6];
sensT = new double[Samp_Pop*6];
string str1 = "ConfidenceDATA_confid6percent.csv";
output1.open(str1.c_str(), ios::app);
Dave = q0[0];
Kave = q0[1];
OmegaAve = q0[2];
m = 0.5;
n = 1.0;
468 readLand(X, Y, Z, initialTime);
a = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, T1, T2, finalTime);
replaceError(Z1, Z);
readLand(X, Y, Z, finalTime);
sigma0 = meanSquareError(Z1,Z,M*N)*sqrt(M*N)/sqrt(M*N - 3);
int nn, mm;
cout << "D = " << q0[0]<< " "<< " sigma = " << sigma0 << " t = 1.9695 (95%)" <<endl;
cout << "K = " << q0[1]<< " "<< " sigma = " << sigma0 << " t = 1.9695 (95%)" <<endl;
cout << "Tc = " << q0[2]<< " "<< " sigma = " << sigma0 << " t = 1.9695 (95%)" <<endl;
double range = 0.06;
478 double dq[3] ;
for(i = 0; i < Samp_Pop; i++)
{
dq[0] = -range+(double)rand() / (double)RAND_MAX*2*range;
dq[1] = -range+(double)rand() / (double)RAND_MAX*2*range;
dq[2] = -range+(double)rand() / (double)RAND_MAX*2*range;

for (j = 0;j<3;j++)
if (dq[j] ==0)
488 {
dq[j] = 1e-6;
}
q[0*Samp_Pop+i] = q0[0]*(1 + dq[0]);

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
q[1*Samp_Pop+i] = q0[1]*(1 + dq[1]);
q[2*Samp_Pop+i] = q0[2]*(1 + dq[2]);

Dave = q[0*Samp_Pop+i] ;
Kave = q[1*Samp_Pop+i] ;
OmegaAve = q[2*Samp_Pop+i] ;
readLand(X, Y, Z, initialTime);
498 a = stepRun(X, Y, Z, Zstar, Q, SLOPE, PHI, dZstar, T1, T2, finalTime);
replaceError(Z1, Z);
readLand(X, Y, Z, finalTime);
sigma[i] = meanSquareError(Z1,Z,M*N)*sqrt(M*N)/sqrt(M*N - 3);
output1 <<q[0*Samp_Pop+i]<<" "<<q[1*Samp_Pop+i]<<" "<<q[2*Samp_Pop+i]<<" "<<sigma[i]<<endl;
cout <<q[0*Samp_Pop+i]<<" "<<q[1*Samp_Pop+i]<<" "<<q[2*Samp_Pop+i]<<" "<<sigma[i]<<endl;
}
sigmaBAR = 0;
for(i = 0; i < Samp_Pop; i++)
{
508 sigmaBAR += (sigma[i]-sigma0)*(sigma[i]-sigma0);
}

sigmaBAR = sqrt(sigmaBAR)/sqrt(Samp_Pop - 1);
cout << "sigmaBAR="<<sigmaBAR<<endl;
for(i = 0; i < Samp_Pop-1; i++)
{
sens[0*Samp_Pop+ i] =0.5*( (sigma[i]-sigma0)/(q[0*Samp_Pop+i]-q0[0]) -
(sigma[i+1]-sigma0)/(q[0*Samp_Pop+i+1]-q0[0]))/(q[0*Samp_Pop+i]-q[1*Samp_Pop+i+1]);
sens[1*Samp_Pop+ i] =0.5*( (sigma[i]-sigma0)/(q[1*Samp_Pop+i]-q0[1]) -
518 (sigma[i+1]-sigma0)/(q[1*Samp_Pop+i+1]-q0[1]))/(q[1*Samp_Pop+i]-q[1*Samp_Pop+i+1]);
sens[2*Samp_Pop+ i] =0.5*( (sigma[i]-sigma0)/(q[2*Samp_Pop+i]-q0[2]) -
(sigma[i+1]-sigma0)/(q[2*Samp_Pop+i+1]-q0[2]))/(q[2*Samp_Pop+i]-q[2*Samp_Pop+i+1]);

sens[3*Samp_Pop+ i] =( (sigma[i]-sigma0)/(q[0*Samp_Pop+i]-q0[0]) -
(sigma[i+1]-sigma0)/(q[0*Samp_Pop+i+1]-q0[0]))/(q[1*Samp_Pop+i]-q[1*Samp_Pop+i+1]);
sens[4*Samp_Pop+ i] =( (sigma[i]-sigma0)/(q[0*Samp_Pop+i]-q0[0]) -
(sigma[i+1]-sigma0)/(q[0*Samp_Pop+i+1]-q0[0]))/(q[2*Samp_Pop+i]-q[2*Samp_Pop+i+1]);
sens[5*Samp_Pop+ i] =( (sigma[i]-sigma0)/(q[1*Samp_Pop+i]-q0[1]) -
528 (sigma[i+1]-sigma0)/(q[1*Samp_Pop+i+1]-q0[1]))/(q[2*Samp_Pop+i]-q[2*Samp_Pop+i+1]);
cout <<i;
for(j = 0;j<6;j++)
cout<<" "<<sens[j*Samp_Pop+ i];
cout <<endl;
}
i = Samp_Pop - 1;
sens[0*Samp_Pop+ i] =0.5*( (sigma[i]-sigma0)/(q[0*Samp_Pop+i]-q0[0]) -
(sigma[i-1]-sigma0)/(q[0*Samp_Pop+i-1]-q0[0]))/(q[0*Samp_Pop+i]-q[0*Samp_Pop+i-1]);
sens[1*Samp_Pop+ i] =0.5*( (sigma[i]-sigma0)/(q[1*Samp_Pop+i]-q0[1]) -
(sigma[i-1]-sigma0)/(q[1*Samp_Pop+i-1]-q0[1]))/(q[1*Samp_Pop+i]-q[1*Samp_Pop+i-1]);
538 sens[2*Samp_Pop+ i] =0.5*( (sigma[i]-sigma0)/(q[2*Samp_Pop+i]-q0[2]) -
(sigma[i-1]-sigma0)/(q[2*Samp_Pop+i-1]-q0[2]))/(q[2*Samp_Pop+i]-q[2*Samp_Pop+i-1]);
sens[3*Samp_Pop+ i] =( (sigma[i]-sigma0)/(q[0*Samp_Pop+i]-q0[0]) -
(sigma[i-1]-sigma0)/(q[0*Samp_Pop+i-1]-q0[0]))/(q[1*Samp_Pop+i]-q[1*Samp_Pop+i-1]);
sens[4*Samp_Pop+ i] =( (sigma[i]-sigma0)/(q[0*Samp_Pop+i]-q0[0]) -
(sigma[i-1]-sigma0)/(q[0*Samp_Pop+i-1]-q0[0]))/(q[2*Samp_Pop+i]-q[2*Samp_Pop+i-1]);
sens[5*Samp_Pop+ i] =( (sigma[i]-sigma0)/(q[1*Samp_Pop+i]-q0[1]) -
(sigma[i-1]-sigma0)/(q[1*Samp_Pop+i-1]-q0[1]))/(q[2*Samp_Pop+i]-q[2*Samp_Pop+i-1]);
transpose(sens, sensT, Samp_Pop, 6);
multiple(sensT, sens, Z, 6, Samp_Pop, Samp_Pop, 6);
548 //for (int i = 0; i<3;i++)
// cout << "matrix:"<< Z[0*3 + i] <<" "<<Z[1*3 + i] <<" "<<Z[2*3 + i] <<endl;
matrixInversion(Z, 6, SLOPE);
cout <<" "<<1.9695*sigmaBAR*sqrt(SLOPE[0*6 + 0]) <<" "<<1.9695*sigmaBAR*sqrt(SLOPE[1*6 + 1]) <<
" "<< 1.9695*sigmaBAR*sqrt(SLOPE[2*6 + 2]) << endl;
output1 <<" "<<1.9695*sigmaBAR*sqrt(SLOPE[0*6 + 0]) <<" "<<1.9695*sigmaBAR*sqrt(SLOPE[1*6 + 1]) <<
" "<< 1.9695*sigmaBAR*sqrt(SLOPE[2*6 + 2]) << endl;
delete [] Z1;
delete [] Z2;
delete [] sens;
558 delete [] sensT;
delete [] q;
return (0);
}
```

Listing C.7 – ‘analysis.cpp’: Network analysis and data extraction

```

#include "header.h"

void functionSet::drainageAREaandLL(double *AREA, double *LL, double *order)
//the drainage area without considering the rainfall
{
    int i, j;
    //cout << "drainageQ() is running" << endl;
8   for(j=0; j<N; j++)
    {
        AREA[j*M + 0] = 1 ;
        LL[j*M + 0] = dx;
        INPUT[j*M + 0] = 0;

        AREA[j*M + M - 1] = 1;
        LL[j*M + M - 1] = dx;
        INPUT[j*M + M - 1] = 0;
    }
18  for(i=0; i<M; i++)
    {
        AREA[0*M + i] = 1 ;
        LL[0*M + i] = dy ;
        INPUT[0*M + i] = 0;

        AREA[(N-1)*M + i] = 1;
        LL[(N-1)*M + i] = dy;
        INPUT[(N-1)*M + i] = 0;
    }
28  for(j=N-2; j>0; j--)
    for(i=1; i<M-1; i++)
    {
        AREA[j*M + i] = 1; // the initial area of each single cell
        INPUT[j*M + i] = 0;

        if(drainTo[j*M + i] == j*M + i + 1 || drainTo[j*M + i] == j*M + i - 1)
            // the initial lenght of each single cell
            LL[j*M + i] = dx;
        else if(drainTo[j*M + i] == (j - 1)*M + i || drainTo[j*M + i] == (j + 1)*M + i)
38         LL[j*M + i] = dy;
        else
            LL[j*M + i] = sqrt(dx*dx + dy*dy);

        if (DIR[j*M + i + 1] == 16)
            INPUT[j*M + i] += 1;

        if (DIR[(j + 1)*M + i + 1] == 8)
            INPUT[j*M + i] += 1;

48         if (DIR[(j + 1)*M + i] == 4)
            INPUT[j*M + i] += 1;

        if (DIR[(j + 1)*M + i - 1] == 2)
            INPUT[j*M + i] += 1;

        if (DIR[j*M + i - 1] == 1)
            INPUT[j*M + i] += 1;

        if (DIR[(j - 1)*M + i - 1] == 128)
58         INPUT[j*M + i] += 1;

        if (DIR[(j - 1)*M + i] == 64)
            INPUT[j*M + i] += 1;

        if (DIR[(j - 1)*M + i + 1] == 32)
            INPUT[j*M + i] += 1;
    }
    int count = 10;
68    int count1;
    int mm, nn;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
int ord = 0;
while (count > 0)
{
    countl=0;

    for(j=N-1; j>=1; j--)
    for(i=0; i<M; i++)
    {
78         if(INPUT[j*M + i] == 0)
        {
            INPUT[j*M + i] = ord;
            countl++;
            AREA[drainTo[j*M + i]] += AREA[j*M + i];
            if(INPUT[drainTo[j*M + i]] == 1)
                LL[drainTo[j*M + i]] += LL[j*M + i];
            INPUT[drainTo[j*M + i]] -= 1;
            INPUT[j*M + i] = 1000;
88        }
    }

    count = countl;
}
countl=0;
for(j=N-2; j>0; j--)
for(i=1; i<M-1; i++)
{
    if(INPUT[j*M + i] != 1000)
98    {
        countl++;
    }
}

for(j=0; j<N; j++)
for(i=0; i<M; i++)
    AREA[j*M + i] *=dx*dy;
}

108 int functionSet::Qcomponents(double *X, double *Y, double *Q, double *Qx, double *Qy)
{
    double d;
    int i, j;
    for(j = 1; j < N - 1 ; j++)
    for(i = 1; i < M - 1; i++)
    {
        d = sqrt((X[drainTo[j*M + i]] - X[j*M + i])*(X[drainTo[j*M + i]] -
X[j*M + i]) + (Y[drainTo[j*M + i]] - Y[j*M + i])*(Y[drainTo[j*M + i]] - Y[j*M + i]));
        Qx[j*M + i] = Q[j*M + i]*(X[drainTo[j*M + i]] - X[j*M + i])/d;
118        Qy[j*M + i] = Q[j*M + i]*(Y[drainTo[j*M + i]] - Y[j*M + i])/d;
    }

    for(j = 1; j < N - 1 ; j++)
    {
        i = 0;
        Qx[j*M + i] = Q[j*M + i];
        Qy[j*M + i] = 0;

        i = M - 1;
        Qx[j*M + i] = -Q[j*M + i];
128        Qy[j*M + i] = 0;
    }

    for(i = 1; i < M - 1 ; i++)
    {
        j = 0;
        Qx[j*M + i] = 0;
        Qy[j*M + i] = Q[j*M + i];

        j = N - 1;
138        Qx[j*M + i] = 0;
        Qy[j*M + i] = -Q[j*M + i];
    }
}
```

```

i = M/2; j = 0;
Qx[j*M + i] = 0;
Qy[j*M + i] = -Q[j*M + i];
Qx[0*M + 0] = (Qx[1*M + 0] + Qx[0*M + 1]) / 2;
Qx[0*M + M - 1] = (Qx[1*M + M - 1] + Qx[0*M + M - 2]) / 2;
Qx[(N - 1)*M + 0] = (Qx[(N - 1)*M + 1] + Qx[(N - 2)*M + 0]) / 2;
Qx[(N - 1)*M + M - 1] = (Qx[(N - 2)*M + M - 1] + Qx[(N - 1)*M + M - 2]) / 2;
148 Qy[0*M + 0] = (Qy[1*M + 0] + Qy[0*M + 1]) / 2;
Qy[0*M + M - 1] = (Qy[1*M + M - 1] + Qy[0*M + M - 2]) / 2;
Qy[(N - 1)*M + 0] = (Qy[(N - 1)*M + 1] + Qy[(N - 2)*M + 0]) / 2;
Qy[(N - 1)*M + M - 1] = (Qy[(N - 2)*M + M - 1] + Qy[(N - 1)*M + M - 2]) / 2;

return 1;
}

int functionSet::calcDrainFrom(double *BV)
// instead of Q, the other parameters can be used (For example
158 //to find the maximum lenghts, the LL parameter can be used.
{
    int i, j;
    // The grid should be filled and the directions be set (filling(),
    //direction())before using this function
    double max = 0;
    int maxPlace = 0;
    int mm, nn;
    for(j=0; j<N; j++)
    {
168     drainFrom[j*M + 0] = j*M + 0;
        drainFrom[j*M + M - 1] = j*M + M - 1;
    }

    for(i=0; i<M; i++)
    {
        drainFrom[0*M + i] = 0*M + i ;
        drainFrom[(N-1)*M + i] = (N-1)*M + i;
    }
    for(j = 1; j<N-1; j++)
178 for(i = 1; i<M-1; i++)
    {
        max = 0;
        for(mm = -1; mm<2 ; mm++)
        for(nn = -1; nn<2 ; nn++)
        {
            if(drainTo[(j + nn)*M + i + mm] == j*M + i)
            {
                if(BV[(j + nn)*M + i + mm] > max)
                {
188                     max = BV[(j + nn)*M + i + mm];
                        maxPlace = (j + nn)*M + i + mm;
                }
            }
        }
        drainFrom[j*M + i] = maxPlace ;
    }
}

int functionSet::mainRiver(double *X, double *Y, double *Z, double *L, double
198 *BV, double BVmin, double *SLOPE, double *Q, double *CURVE, double *LL, double
*PHI, string folder, string fileName) //the main path is determined base on the based value (BV) and its minimum value (BVmin)
{
    int i, j;
    double max = 0;
    int maxPlace;
    int mm, nn;

    prepareFile(folder, fileName); //in prepares the output l for writing
    max = 0;
208 for(i = 1; i<M-1; i++)
    if(BV[2*M + i] > max)
    {
        max = BV[2*M + i];
    }
}

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
maxPlace = 2*M + i;
}
output1 << "X" << " " << "Y" << " " << "Z" << " " << "L" << " " << "LL"
<< " " << "BaseValue" << " " << "SLOPE" << " " << "DISCHARGE" << " "
<< "CURVE" << " " << "filteredCurve1" << " " << "filteredCurve10" << " "
<< "filteredSlope1" << " " << "filteredSlope10" << endl;
218 filter33(CURVE, a, 1); // a, b, c and d are used in the numerical process
filter33(CURVE, b, 10);
filter33(SLOPE, c, 1);
filter33(SLOPE, d, 10);
while(BV[maxPlace]>= BVmin)
{
    output1 << X[maxPlace] << " " << Y[maxPlace] << " " << Z[maxPlace]
    << " " << L[maxPlace] << " " << LL[maxPlace] << " " << BV[maxPlace]
    << " " << SLOPE[maxPlace] << " " << Q[maxPlace] << " " << CURVE[maxPlace]
    << " " << a[maxPlace] << " " << b[maxPlace] << " " << c[maxPlace] << " " << d[maxPlace] << endl;
228 maxPlace = drainFrom[maxPlace];
    //drainFrom defines the input point with the maximum discharge
}

return 0;
}

void functionSet::UpLenght(double *L)
// the length of the points are calculated based on the flow rate
{
238 int i, j;
    for(j=0; j<N; j++)
    {
        L[j*M + 0] = dx ;
        INPUT[j*M + 0] = 0;

        L[j*M + M - 1] = dx;
        INPUT[j*M + M - 1] = 0;
    }
    for(i=0; i<M; i++)
248 {
        L[0*M + i] = dy ;
        INPUT[0*M + i] = 0;

        L[(N-1)*M + i] = dy;
        INPUT[(N-1)*M + i] = 0;
    }
    for(j=N-2; j>0; j--)
    for(i=1; i<M-1; i++)
    {
258 if(drainTo[j*M + i] == j*M + i + 1 || drainTo[j*M + i] == j*M + i - 1)
        L[j*M + i] = dx;
    else if(drainTo[j*M + i] == (j - 1)*M + i || drainTo[j*M + i] == (j - 1)*M + i)
        L[j*M + i] = dy;
    else
        L[j*M + i] = sqrt(dx*dx + dy*dy);
        INPUT[j*M + i] = 0;

    if (DIR[j*M + i + 1] == 16)
        INPUT[j*M + i] += 1;
268 if (DIR[(j + 1)*M + i + 1] == 8)
        INPUT[j*M + i] += 1;

    if (DIR[(j + 1)*M + i] == 4)
        INPUT[j*M + i] += 1;

    if (DIR[(j + 1)*M + i - 1] == 2)
        INPUT[j*M + i] += 1;
278 if (DIR[j*M + i - 1] == 1)
        INPUT[j*M + i] += 1;

    if (DIR[(j - 1)*M + i - 1] == 128)
        INPUT[j*M + i] += 1;
```

```

    if (DIR[(j - 1)*M + i] == 64)
        INPUT[j*M + i] += 1;

    if (DIR[(j - 1)*M + i + 1] == 32)
288     INPUT[j*M + i] += 1;
    }
    int count = 10;
    int count1;
    while (count > 0)
    {
        count1=0;
        for (j=N-2; j>0; j--)
        for (i=1; i<M-1; i++)
        {
298         if (INPUT[drainFrom[j*M + i]] == 0)
            {
                count1++;
                L[j*M + i] += L[drainFrom[j*M + i]];
                INPUT[j*M + i] = 0;
                INPUT[drainFrom[j*M + i]] = 1000;
            }
        }

        count = count1;
308     }
    }

    int functionSet::statistics(double *X, double *Y, double *Z, double *Q, double
    *AREA, double *SLOPE, double *CURVE, double *PHI, double *L, double *LL,
    double percentage, string folder)
    //percentage(0-1) defines the ratio of points to be written
    {
        int i, j;
        prepareFile(folder, "QLAREASlope.dat");
318     output1 << "L(mm)"<<"", " << "LL(mm)"<<"", " <<"Slope(_)" << "", "
        << "Q(mmB/h)" <<"", " << "AREA(mm2)" <<"", " << "power(sqrt(Q)*slope)" << endl;
        for (i = 0; i<percentage*M*N; i++) // it writes a percentage of points: x(%)*MN
        {
            j = int((double)rand()/(double)RAND_MAX*M*N);
            output1 << L[j]<<"", " << LL[j]<<"", " <<SLOPE[j] << "", " << Q[j]
                <<"", " << AREA[j] <<"", " <<sqrt(Q[j])*SLOPE[j]<< endl;
        }
        writeExceedance(Q, PHI, 101, folder, "ExceedanceQ.dat");
        //it writes the exceedance probabilities of a specific parameter
328     writeExceedance(L, PHI, 101, folder, "ExceedanceL.dat");
        writeExceedance(CURVE, PHI, 101, folder, "ExceedanceCURVE.dat");
        writeExceedance(SLOPE, PHI, 101, folder, "ExceedanceSLOPE.dat");
        writeExceedance(AREA, PHI, 101, folder, "ExceedanceAREA.dat");
    }

    void functionSet::writeExceedance(double *paramMain, double *paramTemp, int NumPoints,
    string folderName, string fileName)
    //it writes the exceedance probabilities of a specific parameter
338 {
        int i, j;
        double max, min;
        int counter, k;
        prepareFile( folderName, fileName);

        for (j = 0; j<N; j++)
        for (i = 0; i<M; i++)
        {
            if (paramMain[j*M + i] > max)
348             max = paramMain[j*M + i];
            else if (paramMain[j*M + i] < min)
                min = paramMain[j*M + i];
        }
        paramTemp[0] = min;
        output1 << min <<" " << 1.0 << "\n";
    }

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
for(k = 1; k<NumPoints; k++)
{
    counter = 0;
    paramTemp[k] = min + (double)rand() / (double)RAND_MAX * (max-min);
358
    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
    {
        if(paramMain[j*M + i] >= paramTemp[k])
            counter++;
    }
    output1 << paramTemp[k] << " " << counter / double(M*N) << endl;
}

368 void functionSet::prepareFile(string folderName, string fileName)
//it prepares the output 1 for writing
{
    output1.close();
    folderName += fileName;
    remove(folderName.c_str());
    output1.open(folderName.c_str(), ios::app);
}

void functionSet::filter33(double *param, double *filtered, int filtTime)
378 //it filters the data by a 3 by 3 cells
{
    int i, j;
    int nml, nnl, ll;
    double temp;
    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
        filtered[j*M + i] = param[j*M + i];

    ll=0;
388 while(ll<filtTime)
    {
        for(j = 1; j<N-1; j++)
        for(i = 1; i<M-1; i++)
        {
            temp = 0;
            for(nml = -1; nml<2 ; nml++)
            for(nnl = -1; nnl<2 ; nnl++)
            {
398                 temp += filtered[(j + nnl)*M + i + nml];
            }
            filtered[j*M + i] = temp/9.0;
        }
        ll++;
    }

    double functionSet::filterCELL(double *param, int i, int j, int n)
//it filters param[i,j] by an n by n cells
{
408     int nml, nnl;
    double temp = 0;
    for(nml = -n/2; nml<n/2+1 ; nml++)
    for(nnl = -n/2; nnl<n/2+1 ; nnl++)
    {
        temp += param[(j + nnl)*M + i + nml];
    }
    return temp/double(n*n);
}

418 double functionSet::filternn(double *param, double *filtered, int n)
//it filters param by an n by n cells
{
    int i, j;
    int nml, nnl, ll;
    double temp;
```

```

    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
        filtered[j*M + i] = param[j*M + i];
428
    for(j = 1; j<N-1; j++)
    {
        if(j<n/2)
        for(i = 1; i<M-1; i++)
            filtered[j*M + i] = filterCELL(filtered, i, j, 2*j + 1);

        else if(j>N-n/2 - 1)
        for(i = 1; i<M-1; i++)
            filtered[j*M + i] = filterCELL(filtered, i, j, 2*(-j + N - 1)+1);
438
        else
        {
            for(i = 1; i<M-1; i++)
            {
                if(i<n/2)
                    filtered[j*M + i] = filterCELL(filtered, i, j, 2*i+1);
                else if(i>M-n/2 - 1)
                    filtered[j*M + i] = filterCELL(filtered, i, j, 2*(-i + M - 1)+1);
                else
                    filtered[j*M + i] = filterCELL(filtered, i, j, n);
448
            }
        }
    }
    for(j=0; j<N; j++)
    {
        filtered[j*M + 0] = filtered[j*M + 1]; filtered[j*M + M - 1] = filtered[j*M + M - 2];
    }

    for(i=0; i<M; i++)
    {
458
        filtered[(N-1)*M + i] = filtered[(N-2)*M + i]; filtered[0*M + i] = filtered[1*M + i];
    }
}

void functionSet::filterNetwork(double *Z1, double *Z2, double *Y, double *Qopt)
//it filters the value of Z2(experiment) according to the Z1(model) to have
//the most match between the model and experiment
{
    int i, j;
    int optFilter;
468
    double minSQR, optMinSQR;
    double *Zstar1, *PHI1, *Q1, *Q2;
    Zstar1 = new double[M*N];
    PHI1 = new double[M*N];
    Q1 = new double[M*N];
    Q2 = new double[M*N];
    for(i = 1; i < M*N; i++)
    {
        Zstar1[i] = 0;
        PHI1[i] = 0;
478
        Q1[i] = 0;
        Q2[i] = 0;
    }
    replaceError(Zstar1, Z1);
    filling(Zstar1, PHI1, Y);
    direction(Zstar1, PHI1, Y);
    drainageQ(Q1);
    replaceError(Zstar1, Z2);
    filling(Zstar1, PHI1, Y);
    direction(Zstar1, PHI1, Y);
488
    drainageQ(Q2);
    minSQR = 0;
    for(i = 0; i < M*N; i++)
    {
        minSQR += (log10(Q1[i]) - log10(Q2[i]))*(log10(Q1[i]) - log10(Q2[i]));
    }
    optFilter = 0;
    optMinSQR = minSQR;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
replaceError(Qopt, Q2);

498 for(i = 1; i < 200; i++)
{
    filter33(Z2, Zstar1, i);
    filling(Zstar1, PH11, Y);
    direction(Zstar1, PH11, Y);
    drainageQ(Q2);
    minSQR = 0;
    for(j = 0; j < M*N; j++)
    {
508 minSQR += (log10(Q1[j]+ 1) - log10(Q2[j]+1))*(log10(Q1[j]+1) - log10(Q2[j]+1));
    }
    cout <<i <<" " <<minSQR <<endl;
    if(minSQR < optMinSQR)
    {
        optFilter = i;
        optMinSQR = minSQR;
        replaceError(Qopt, Q2);
    }
}

518 cout << "optimum filter = " << optFilter <<endl;
}

void functionSet::findMinMax(double *param, double *min, double *max)
{
    int i, j;
    *min = 1e15;
    *max = -1e15;
    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
528 {
        if(param[j*M + i] > *max)
            *max = param[j*M + i];
        else if(param[j*M + i] < *min)
            *min = param[j*M + i];
    }
}

void functionSet::normalize(double *X)
{
538 int i, j;
    double ave = average(X);
    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
        X[j*M + i] = X[j*M + i]/ave;
}

double functionSet::average(double *X)
{
    int i, j;
548 double ave = 0;
    for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)
        ave += X[j*M + i];
    return(ave/double(M*N));
}

double functionSet::RMS(double *X, int size)
//https://brenocon.com/blog/2012/03/cosine-similarity-
//pearson-correlation-and-ols-coefficients/
558 {
    int i, j;
    double aveX, RMS;
    aveX = average(X);
    RMS = 0;

    for(j = 0; j<size; j++)
    {
        RMS += (X[j] - aveX)*(X[j] - aveX);
    }
}
```

```

}
568 return sqrt(RMS);
}

double functionSet::meanSquareError(double *X, double *Y, int size)
//https://brenocon.com/blog/2012/03/
//cosine-similarity-pearson-correlation-and-ols-coefficients/
{
    int i, j;
    double RMS;
578 RMS = 0;

    for(j = 0; j<size; j++)
    {
        RMS += (X[j] - Y[j])*(X[j] - Y[j]);
    }

    return sqrt(RMS/size);
}

588 double functionSet::correlation(double *X, double *Y, int size)
//https://brenocon.com/blog/2012/03/cosine-similarity
//pearson-correlation-and-ols-coefficients/
{
    int i, j;
    double aveX, aveY, RMSx, RMSy, corr ;
    aveX = average(X);
    aveY = average(Y);
    RMSx = RMS(X, size);
    RMSy = RMS(Y, size);
598 corr = 0;

    for(j = 0; j<size; j++)
    {
        corr += (X[j] - aveX)*(Y[j] - aveY);
    }

    return corr/(RMSx * RMSy);
}

608 void functionSet::L_down(double *downL)
{
    int i, j, newPlace;
    for(j=1; j<N-1; j++)
    for(i=1; i<M-1; i++)
    {
        downL[j*M + i] = 0;
        newPlace = drainTo[j*M + i];
        while (newPlace != -1)
        {
818 if(newPlace == j*M + i + 1 || newPlace == j*M + i - 1)
// the initial lenght of each single cell
            downL[j*M + i] += dx;
        else if(newPlace == (j - 1)*M + i || newPlace == (j + 1)*M + i)
            downL[j*M + i] += dy;
        else
            downL[j*M + i] += sqrt(dx*dx + dy*dy);
        cout<<j*M + i<<" "<<newPlace<<endl;
        newPlace = drainTo[newPlace];
628 }
    }
    cout<<drainTo[newPlace]<<endl;
}

```

Listing C.8 – ‘matrixOperation.cpp’: Basic matrix operations

```

#include "header.h"
int matrixOperations::transpose(double *A, double *B, int row, int col)

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
{
    int i,j;
    double *temp;
    temp = new double[row*col];
    for(i=0; i<row; i++)
9   for(j=0; j<col; j++)
    {
        temp[i*col + j] = A[j*row + i];
    }
    for(i=0; i<row; i++)
    for(j=0; j<col; j++)
    {
        B[i*col + j] = temp[i*col + j];
    }
    delete [] temp;
19  return 0;
}

int matrixOperations::multiple(double *A, double *B, double *C, int Ar, int Ac, int Br, int Bc)
{
    int i,j,k;
    /* If column of first matrix is not equal to row of second matrix, asking user to enter the size of matrix again. */
    while (Ac!=Br)
    {
        cout << "Error! column of first matrix not equal to row of second.";
29     return 1;
    }

    /* Initializing elements of matrix mult to 0.*/
    for(i=0; i<Ar; ++i)
    for(j=0; j<Bc; ++j)
    {
        C[j*Ar + i] = 0;
    }

39 /* Multiplying matrix a and b and storing in array mult. */
    for(i=0; i<Ar; ++i)
    for(j=0; j<Bc; ++j)
    for(k=0; k<Ac; ++k)
    {

        C[j*Ar + i] += A[k*Ar + i]*B[j*Br + k];
    }
    return 0;
}

49 /// calculate the cofactor of element (row,col)
int matrixOperations::getMinor(double **src, double **dest, int row, int col, int order)
{
    // indicate which col and row is being copied to dest
    int colCount=0,rowCount=0;

    for(int i = 0; i < order; i++ )
    {
        if( i != row )
        {
59             colCount = 0;
            for(int j = 0; j < order; j++ )
            {
                // when j is not the element
                if( j != col )
                {
                    dest[rowCount][colCount] = src[i][j];
                    colCount++;
                }
            }
69             rowCount++;
        }
    }

    return 1;
}
```

```

}

// Calculate the determinant recursively.
double matrixOperations::calcDeterminant( double **mat, int order)
{
79 // order must be >= 0
// stop the recursion when matrix is a single element
if( order == 1 )
    return mat[0][0];

// the determinant value
double det = 0;

// allocate the cofactor matrix
double **minor;
89 minor = new double*[order-1];
for(int i=0;i<order-1;i++)
    minor[i] = new double[order-1];

for(int i = 0; i < order; i++ )
{
    // get minor of element (0,i)
    getMinor( mat, minor, 0, i , order);
    // the recursion is here!
    det += pow( -1.0, i ) * mat[0][i] * calcDeterminant( minor,order-1 );
99 }

// release memory
for(int i=0;i<order-1;i++)
    delete [] minor[i];
delete [] minor;

return det;
}

109 // matrix inversion
// the result is put in Y
void matrixOperations::matrixInversion(double *A, int order, double *Y)
{
    // memory allocation
    double *temp1 = new double[order*order];
    double **AA = new double*[order];
    for(int i=0;i<order;i++)
        AA[i] = temp1+(i*(order));

119 for(int j=0;j<order;j++)
    for(int i=0;i<order;i++)
        AA[i][j] = A[j*order + i];

    // get the determinant of a
    //cout<<calcDeterminant(AA,order)<<endl;
    double det = 1.0/calcDeterminant(AA,order);

    // memory allocation
    double *temp = new double[(order-1)*(order-1)];
129 double **minor = new double*[order-1];
    for(int i=0;i<order-1;i++)
        minor[i] = temp+(i*(order-1));

    for(int j=0;j<order;j++)
    {
        for(int i=0;i<order;i++)
        {
            // get the co-factor (matrix) of A(j,i)
            getMinor(AA,minor,i,j,order);
            Y[j*order + i] = det*calcDeterminant(minor,order-1);
139 if( (i+j)%2 == 1)
                Y[j*order + i] = -Y[j*order + i];
        }
    }
}

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
// release memory
delete [] minor[0];
delete [] minor;
}
```

Listing C.9 – ‘K_means.cpp’: 1D and 2D K-means algorithm

```
1 #include "header.h"
void functionSet::Ktest(double *X, double *Y, double *center, int maxK,
                      string folderName, string fileName)
{
    int i, j, k, myK, n, m;
    int *group, *groupPop;
    group = new int [M*N];
    groupPop = new int [maxK];
    double error, min, distance, aa;
    double *bb;
11 bb = new double [maxK];
    output2.close();
    folderName += fileName;
    remove(folderName.c_str());
    output2.open(folderName.c_str(), ios::app);
    for(myK=2; myK<maxK; myK+=1)
    {
        error = 0;
        Kmean(X, Y, center, M*N, myK, folderName, "/test.txt");
        for(k = 0; k < myK; k++)
21 {
            groupPop[k] = 0;
        }
        for(j = 0; j<N; j++)
        for(i = 0; i<M; i++)
        {
            min = 1e15;
            for(k = 0; k < myK; k++)
            {
                //distance = sqrt( (X[j*M + i] - center[0*myK + k])*(X[j*M + i]
31 // - center[0*myK + k]) + (Y[j*M + i] - center[1*myK + k])
                //(Y[j*M + i] - center[1*myK + k]) );
                distance = sqrt( (X[j*M + i] - center[0*myK + k])
                *(X[j*M + i] - center[0*myK + k]) )/1e15;

                if(distance < min)
                {
                    group[j*M + i] = k;
                    min = distance;
                }
41 }
            for(k = 0; k < myK; k++)
            {
                if(group[j*M + i] == k)
                    groupPop[k] += 1;
            }
            //cout << group[j*M + i] << endl;
        }
51 for(j = 0; j<N; j++)
        for(i = 0; i<M; i++)
        {
            for(k = 0; k < myK; k++)
            if(group[j*M + i] == k)
                error += sqrt( (X[j*M + i] - center[0*myK + k])*(X[j*M + i] - center[0*myK + k]) );
            //error += sqrt( (X[j*M + i] - center[0*myK + k])*(X[j*M + i] -
            //center[0*myK + k]) + (Y[j*M + i] - center[1*myK + k])*(Y[j*M + i] - center[1*myK + k]) );
        }
61 output2 << myK << " " << error/(M*N) << " ";
}
```

```

error = 0; //to save the summation of S's
for(j = 0; j<N; j++)
for(i = 0; i<M; i++)
{
    a = 0;
    for(k = 0; k < myK; k++)
        bb[k] = 0;

71     for(n = 0; n<N; n++) // the distance from each point is added to b[k]'s
        for(m = 0; m<M; m++)
        {
            //bb[group[n*M + m]] += sqrt( (X[j*M + i] - X[n*M + m])*(X[j*M + i]
            // - X[n*M + m]) + (Y[j*M + i] - Y[n*M + m])*(Y[j*M + i] - Y[n*M + m]) );
            bb[group[n*M + m]] += sqrt( (X[j*M + i] - X[n*M + m])*(X[j*M + i] - X[n*M + m]) );

        }
    for(k = 0; k < myK; k++)
81     bb[k] /= groupPop[k];

    aa = bb[group[j*M + i]];

    min = 1e20;
    for(k = 0; k < myK && k != group[j*M + i]; k++)
        if(bb[k] < min)
        {
            min = bb[k];
        }

91     if(aa < min)
        error += 1- aa/min;
    else if ( aa > min)
        error += min/aa - 1;
}

output2 << myK <<"    "<< error/(M*N) <<endl;
}

101 delete[] group;
delete[] bb;
delete[] groupPop;
}

void functionSet::Kmean(double *XX, double *YY, double *center, int NumPoint,
int myK, string folderName, string fileName)
//Attention:the distance can be changed in X, Y or both directions, also be
//careful about the calculation of variance (just variance of y value has been calculated.
{
    int i,j;
111 double distance, *groupVar, *centerOLD, *X,*Y, min,max;
    int *group, *groupPop,k;
    group = new int [NumPoint];
    X = new double [NumPoint];
    Y = new double [NumPoint];
    centerOLD = new double [2*myK];
    groupPop = new int [myK];
    groupVar = new double [myK];
    for(j = 0; j<NumPoint; j++)
    {
121     X[j] = log(XX[j])/log(10.0); //log(XX[j])/log(10.0);
        Y[j] = log(YY[j])/log(10.0); //log(YY[j])/log(10.0);
    }
    min = 1e15;
    max = -1e15;
    for(j = 0; j<NumPoint; j++)
    {
        if(X[j] > max)
            max = X[j];
        else if(X[j] < min)
131     min = X[j];
    }
    /* for(j = 0; j<N; j++)
    for(i = 0; i<M; i++)

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
{
    if (Y[j*M + i] <= 1e-15)
        Y[j*M + i] = 1e-15;
    if (X[j*M + i] <= 1e-15)
        X[j*M + i] = 1e-15;
    X[j*M + i] = log(X[j*M + i]) / log(10.0);
    Y[j*M + i] = log(Y[j*M + i]) / log(10.0);
    cout << X[j*M + i] << " " << Y[j*M + i] << endl;
}*/
for(k=0; k<myK; k++)// initial group centers
{
    center[0*myK + k] = min + (max-min)/myK*k; //uniform distribution along X
    i = int((double)rand()/(double)RAND_MAX*NumPoint);
    center[1*myK + k] = Y[i];
}
double ok = 0;
151 while(ok < myK - 1)
{
    ok = 0;
    for(j = 0; j<NumPoint; j++)
    {
        min = 1e15;
        for(k = 0; k < myK; k++)
        {
            //distance = abs( (X[j*M + i] - center[0*myK + k])*(X[j*M + i]
            // - center[0*myK + k]) + (Y[j*M + i] - center[1*myK + k])*(Y[j*M + i] - center[1*myK + k]) )/1e15;
161 distance = sqrt( (X[j] - center[0*myK + k])*(X[j] - center[0*myK + k]) + (Y[j] - center[1*myK + k])*(Y[j] - center[1*myK + k]) )/1e15;

            if(distance < min)
            {
                group[j] = k;
                min = distance;
            }
        }
        //cout << group[j*M + i] << endl;
    }
171 for(k = 0; k < myK; k++)
{
    centerOLD[0*myK + k] = center[0*myK + k];
    centerOLD[1*myK + k] = center[1*myK + k];
    center[0*myK + k] = 0;
    center[1*myK + k] = 0;
    groupPop[k] = 0;
    for(j = 0; j<NumPoint; j++)
    {
        if(group[j] == k)
181 {
            center[0*myK + k] += X[j];
            center[1*myK + k] += Y[j];
            groupPop[k]++;
        }
    }

    if(groupPop[k] >= 1)
    {
        center[0*myK + k] /= double(groupPop[k]);
        center[1*myK + k] /= double(groupPop[k]);
191 }
    else
    {
        i = int((double)rand()/(double)RAND_MAX*M*N);
        center[0*myK + k] = X[i];
        center[1*myK + k] = Y[i];
    }

201 //cout << groupPop<<" " << center[0*myK + k] << " " << center[1*myK + k] << endl;
}

for(k = 0; k < myK; k++)
    if (centerOLD[1*k] == center[0*myK + k] || centerOLD[2*k] == center[1*myK + k])
```



```

    ok++;
    cout << ok << endl;

}

211 for(k = 0; k < myK; k++)
{
    groupVar[k] = 0;
    for(j = 0; j<NumPoint; j++)
    {
        if(group[j] == k)
            groupVar[k] += (center[1*myK + k] - Y[j])*
                (center[1*myK + k] - Y[j])/double(groupPop[k]);
    }
221 }
prepareFile(folderName, fileName);
output1 << "Xmin" << ", " << "Ymin" << ", " << "groupPop" << ", " << "Variance" << endl;
for(int k = 0; k < myK; k++)
    output1 << pow(10, center[0*myK + k]) << ", " << pow(10.0, center[1*myK + k])
    << ", " << groupPop[k] << ", " << groupVar[k] << endl;
delete [] group;
delete [] groupPop;
delete [] centerOLD;
delete [] groupVar;
231 }

```

Listing C.10 – ‘LANDserial.cpp’: Coupling the model with multi-objective Borg optimization code

```

/* Copyright 2012–2014 The Pennsylvania State University
 *
 * This software was written by David Hadka and others.
 *
 * The use, modification and distribution of this software is governed by the
 * The Pennsylvania State University Research and Educational Use License.
 * You should have received a copy of this license along with this program.
 * If not, contact <dmh309@psu.edu>.
 */
10 #include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "borg.h"
#include "header.h"

int nvars = 4;
int nobjs = 2;

// All optimization problems must define a function similar to the one below.
20 // This function is responsible for reading the decision variables and
// evaluating the problem. The resulting objectives and constraints (if any)
// are saved to the objs and consts arguments, respectively. Note that
// all objectives MUST be minimized. For constraints, set the constraint to
// 0 if the constraint is satisfied; any non-zero constraint value is
// considered a constraint violation.
void LAND(double* vars, double* objs, double* consts) {
    int i, j;
    //cout << " in j1" << endl;
    double *X,*Y,*Z,*Q,*SLOPE,*dZstar,*Zstar,*PHI;
30 X = new double[M*N];
Y = new double[M*N];
Z = new double[M*N];
Q = new double[M*N];
SLOPE = new double[M*N];
dZstar = new double[M*N];
Zstar = new double[M*N];
PHI = new double[M*N];

functionSet *MCH;

```

Appendix C. C++ scripts: Landscape evolution model and optimization tools

```
40 MCH = new functionSet [1];

MCH[0].Dave = vars[0];
MCH[0].Kave = vars[1];
MCH[0].m = 0.5;
MCH[0].n = 1.0;
MCH[0].OmegaAve = vars[2];
MCH[0].noiseAve = vars[3];
MCH[0].E = 0;

50 MCH[0].allocate(X,Y,Z,Q,dZstar, Zstar,PHI);
string addressOPT = "results.dat";
ofstream output;
objectives objFun;
output.open(addressOPT.c_str(),ios::app);

char path1[1024];
string path;
size_t size;
if (getcwd(path1, sizeof(path1)) != NULL)
60 path = path1;
else
    perror("getcwd() error");

//cout<<"\n current Path"<<path;
//path = path.substr(0,path.size()-6); // take the "/model" part

string ZinitialTime = path+"input/00h15min_8mm.dat"; //the input file
string ZfinalTime = path+"input/08h00min_8mm.dat"; //the input file
MCH[0].defineRain(path+"input/Rain8mm.dat",1); //0:uniform, 1:variable

70 MCH[0].varD = 0;
string filename[] = {path+"input/00h15min_8mm.dat", path+"input/00h30min_8mm.dat",
    path+"input/01h00min_8mm.dat", path+"input/02h00min_8mm.dat",
    path+"input/04h00min_8mm.dat", path+"input/08h00min_8mm.dat",
    path+"input/16h00min_8mm.dat"};

MCH[0].temporalBCset(X, Y, Z,filename); // captures the BCs from the scans
//objs[0] = abs(vars[0] - 8000)*abs(vars[1] - 0.15)*abs(vars[2] - 50);

80 //objs[1] = MCH[0].objective1(X, Y, Z, Zstar, Q, SLOPE,PHI, dZstar, ZinitialTime,ZfinalTime,filename);
objFun = MCH[0].multiObjective_Zerror_discharge(X, Y, Z, Zstar, Q, SLOPE,PHI,
    dZstar, ZinitialTime,ZfinalTime,filename);

objs[0] = objFun.a;
objs[1] = objFun.b;
//objs[2] = objFun.c;

int nn;
for(nn = 0; nn < nvars; nn++)
    cout << " "<< vars[nn];
90 for(nn = 0; nn < nobjs; nn++)
    cout << " "<<objs[nn];

cout << " "<<endl;

for(nn = 0; nn < nvars; nn++)
    output << " "<< vars[nn];

for(nn = 0; nn < nobjs; nn++)
    output << " "<<objs[nn];
100 output << " "<<endl;

//MCH[0].unAllocate(X,Y,Z,Q,SLOPE,dZstar, Zstar,PHI);
//cout << " inja2" <<endl;
delete [] MCH;

110 /*
```

```

    objs[0] = (vars[0] - 3500)*(vars[0] - 3500);
    int nn;
    for(nn = 0; nn < nvars; nn++)
        cout << " "<< vars[nn];
    cout << " "<<objs[0]<<endl;
    */
}

120 int main(int argc, char* argv[]) {
    remove("results.dat");
    double min[3];
    min[0] = 10000.0 ;//D,
    min[1] = 0.25;//K,
    min[2] = 10;//Omega
    min[3] = 0;//rndCoef

    double max[3];
130 max[0] = 21000+1e-6;//D
    max[1] = 0.5+1e-6;//K,
    max[2] = 100.0 +1e-6;
    max[3] = 600.0;//rndCoef

    double epsilon[8];
    epsilon[0] = 0.01;
    epsilon[1] = 0.01;
    epsilon[2] = 0.01;
140 epsilon[3] = 0.01;
    epsilon[4] = 0.01;
    epsilon[5] = 0.01;
    epsilon[6] = 0.01;

    int ii;

    // Create the DTLZ2 problem, defining the number of decision variables ,
    // objectives and constraints. The last argument, dtlz2, references
    // the function that evaluates the DTLZ2 problem.
150 BORG_Problem problem = BORG_Problem_create(nvars, nobjs, 0, LAND);

    // Set the lower and upper bounds for each decision variable.

    for (ii=0; ii<nvars; ii++) {
        BORG_Problem_set_bounds(problem, ii, min[ii], max[ii]);
    }

    // Set the epsilon values used by the Borg MOEA. Epsilons define the
    // problem resolution, which controls how many Pareto optimal solutions
160 // are generated and how far apart they are spaced.
    for (ii=0; ii<nobjs; ii++) {
        BORG_Problem_set_epsilon(problem, ii, epsilon[ii]);
    }

    // Run the Borg MOEA on the DTLZ2 problem for 1 million function
    // evaluations.
    BORG_Archive result = BORG_Algorithm_run(problem, 200);

    // Print the Pareto optimal solutions.
170 BORG_Archive_print(result, stdout);

    // Free any allocated memory.
    BORG_Archive_destroy(result);
    BORG_Problem_destroy(problem);
    return EXIT_SUCCESS;
}

```


D Selected Python scripts: Data analysis and visualization

The files with a User's Guide are available at:

<https://github.com/mcheraghi/Landscape-Evolution-Model.git>.

Listing D.1 – Exceedance probability of length

```
import matplotlib
import numpy as np
3 from numpy import sqrt, sin, cos, pi
  from numpy import genfromtxt

import matplotlib.cm as cm
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
import csv

from mpl_toolkits.axes_grid1 import ImageGrid
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
13 from numpy.random import uniform, seed
  from mpl_toolkits.axes_grid1 import make_axes_locatable
  from matplotlib.colors import BoundaryNorm
  from matplotlib.ticker import MaxNLocator
  from mpl_toolkits.mplot3d import Axes3D
  from matplotlib import cbook
  from matplotlib.colors import LightSource
  from math import log10
  matplotlib.rcParams['axes.unicode_minus'] = False
  import os
23 import matplotlib.gridspec as gridspec
  from matplotlib import colors, ticker, cm
  from matplotlib.colors import LogNorm
  from sklearn.neighbors import KernelDensity
  from scipy import signal
  from scipy.fftpack import fft, ifft
  import pylab as py
  import scipy.fftpack
  import timeit

33 #title_font = {'fontname':'Times', 'size':'10', 'color':'black', 'weight':'normal',
    # 'verticalalignment':'bottom'} # Bottom vertical alignment for more space
  axis_font = {'fontname':'Times', 'size':'12'}
  #M = 121 #exp8: 241(4mm), 121(8mm)      exp 7:123 (8 mm)//244 (4 mm)
  #N = 239 #exp8: 477(4mm), 239(8mm)      exp 7:233 (8 mm)//464 (4 mm)
  matplotlib.rcParams.update({'font.size': 12})
M = 121    #241    #121
```

Appendix D. Selected Python scripts: Data analysis and visualization

```
N = 239 #477 #239

fileGroup1 = ['L_exp.csv', 'L_model.csv']

43 #-----Exceedance Probability function-----

def Exceedance1(q):
    qlog = np.log10(q)
    minq = min(qlog)
    maxq = max(qlog)
    dqbin = (3 - log10(8))/100.0
    #NOTE: we put 1^3 as the max to be consistent for all model data

53 qselect = [minq + dqbin*i1 for i1 in range(0,101)]
    ExProbq = np.ones(101)

    for k in range(0, 101):
        for j in range(0, len(qlog)):
            if qlog[j] >= qselect[k]:
                ExProbq[k] = ExProbq[k] + 1

    qselect = [pow(10.0,nn) for nn in qselect]
    return qselect, ExProbq/len(q)

63 #-----Exceedance for the model-----

def Exceedance2(qselect, q):
    qlog = np.log10(q)
    qselect = np.log10(qselect)
    ExProbq = np.ones(101)

    for k in range(0, 101):
        for j in range(0, len(qlog)):
73         if qlog[j] >= qselect[k]:
            ExProbq[k] = ExProbq[k] + 1
    return ExProbq/len(q)

#-----Exceedance probability of all experiments-----

def Exceed_exp():
    expl=np.empty((101,6))
    error = 0
    exp = genfromtxt('L_exp.csv', delimiter=',')
    for k in range(1, 6):
83     expl[:,0], expl[:,k] = Exceedance1(exp[:,k-1])

    mat = np.matrix(expl)
    with open('Lexceed_exp.dat', 'wb') as f:
        for line in mat:
            np.savetxt(f, line, delimiter=',')

def rms():
    modell=np.empty((101,6))
    expl = genfromtxt('Lexceed_exp.dat', delimiter=',')
    #the first column is the selected q and the rest are the exceedance probabilities
93    model = genfromtxt('L_model.csv', delimiter=',')

    error = 0
    for k in range(0, 5):
        modell[:,k] = Exceedance2(expl[:,0], model[:,k])
        error = error+np.sqrt(((modell[:,k] - expl[:,k+1]) ** 2.0).mean())

    target = open('L_RMS.dat', 'w') #put the desitination address
    target.write(str(10**error))
```

Listing D.2 – Data analysis and visualization

```
import matplotlib
import numpy as np
import matplotlib.cm as cm
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
```

```

from scipy.fftpack import fft, ifft
7 import pandas as pd
import scipy.stats as st
import statsmodels as sm
import csv
import math
import pylab as py
import scipy.fftpack
import warnings
import operator
from numpy import genfromtxt
17 from mpl_toolkits.axes_grid1 import ImageGrid
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from numpy.random import uniform, seed
from mpl_toolkits.axes_grid1 import make_axes_locatable
from matplotlib.colors import BoundaryNorm
from matplotlib.ticker import MaxNLocator
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cbook
from matplotlib.colors import LightSource
from math import log10
27 matplotlib.rcParams['axes.unicode_minus'] = False
import os
import matplotlib.gridspec as gridspec
from matplotlib import colors, ticker, cm
from matplotlib.colors import LogNorm
#title_font = {'fontname':'Times', 'size':'10', 'color':'black', 'weight':'normal',
#              # 'verticalalignment':'bottom'} # Bottom vertical alignment for more space
plt.rcParams["font.family"] = 'serif'
axis_font = {'fontname':'Times', 'size':'12'}
#M = 121 #exp8: 241(4nm), 121(8nm) exp 7:123 (8 nm)//244 (4 nm)
37 #N = 239 #exp8: 477(4nm), 239(8nm) exp 7:233 (8 nm)//464 (4 nm)
matplotlib.rcParams.update({'font.size': 12})
M = 121
N = 239
my_data=np.empty((10,M*N+1,11))
my_data1=np.empty((10,M*N+1,11))
my_data2=np.empty((10,M*N+1,11))
my_data3=np.empty((10,M*N+1,11))

inFile = os.path.dirname(os.path.realpath(__file__))
47 inFile= inFile
inFile = ".join(inFile.rsplit('/script'))

outFile= inFile
outFile = ".join(outFile.rsplit('/script'))
outFile = outFile + '/plots/chap4_'

f_cr =7
fileName0 = [outFile+"/outputData/XYZ_EXP/XYZ00",
outFile+"/outputData/XYZ_EXP/XYZ01",
57 outFile+"/outputData/XYZ_EXP/XYZ02",
outFile+"/outputData/XYZ_EXP/XYZ03",
outFile+"/outputData/XYZ_EXP/XYZ04",
outFile+"/outputData/XYZ_EXP/XYZ05",
outFile+"/outputData/XYZ_EXP/XYZ06"]

fileName1 = [outFile+"/outputData/XYZ_model1/XYZ00",
outFile+"/outputData/XYZ_model1/XYZ01",
outFile+"/outputData/XYZ_model1/XYZ02",
outFile+"/outputData/XYZ_model1/XYZ03",
67 outFile+"/outputData/XYZ_model1/XYZ04",
outFile+"/outputData/XYZ_model1/XYZ05",
outFile+"/outputData/XYZ_model1/XYZ06"]
fileName2 = [outFile+"/outputData/XYZ_model2/XYZ00",
outFile+"/outputData/XYZ_model2/XYZ01",
outFile+"/outputData/XYZ_model2/XYZ02",
outFile+"/outputData/XYZ_model2/XYZ03",
outFile+"/outputData/XYZ_model2/XYZ04",
outFile+"/outputData/XYZ_model2/XYZ05",
outFile+"/outputData/XYZ_model2/XYZ06"]

```

Appendix D. Selected Python scripts: Data analysis and visualization

```
77 my_data[0,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA00..csv', delimiter=',')
my_data[1,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA01..csv', delimiter=',')
my_data[2,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA02..csv', delimiter=',')
my_data[3,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA03..csv', delimiter=',')
my_data[4,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA04..csv', delimiter=',')
my_data[5,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA05..csv', delimiter=',')
my_data[6,:,:) = genfromtxt(inFile+'data/experiment/statistical_Analysis/original/results/DATA06..csv', delimiter=',')

my_data1[0,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA00..csv', delimiter=',')
87 my_data1[1,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA01..csv', delimiter=',')
my_data1[2,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA02..csv', delimiter=',')
my_data1[3,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA03..csv', delimiter=',')
my_data1[4,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA04..csv', delimiter=',')
my_data1[5,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA05..csv', delimiter=',')
my_data1[6,:,:) = genfromtxt(inFile+'data/model/smooth/RUNS/optimum/uniformD/results/DATA06..csv', delimiter=',')

def Exceedance(q):
    qlong = np.log10(q + 1)
    minq = min(qlong)
97 maxq = max(qlong)
    dqbin = (max(qlong) - min(qlong))/100.0

    qselect = [minq + dqbin*i1 for i1 in range(0,101)]
    ExProbq = np.ones(101)

    for k in range(0, 101):
        for j in range(0, len(qlong)):
            if qlong[j] >= qselect[k]:
                ExProbq[k] = ExProbq[k] + 1
107

    qselect = [pow(10.0,nn) for nn in qselect]
    return qselect, ExProbq/len(q)

def linearPartFit(qselect, ExProbq, Min, Max):

    for k in range(0, len(qselect)):
        if qselect[k] > Min:
            iMin = k
117 break

    for k in range(0, len(qselect)):
        if qselect[k] > Max:
            iMax = k
        break

    q = [np.log10(i1) for i1 in qselect]
    ExPr = [np.log10(i1) for i1 in ExProbq]
    m, n = np.polyfit(q[iMin:iMax], ExPr[iMin:iMax], 1)
127 return m, n

#-----calculating the avsegae fft magnitude-----
def processing(my_data, i, fileName, ax1, ax2, color, legText, markerType, markevery):

    x = my_data[i,1:,0]
    y = my_data[i,1:,1]
    z = abs(my_data[i,1:,2])
    X = np.reshape(x, (M,N))
137 Y = np.reshape(y, (M,N))
    Z = np.reshape(z, (M,N))
    X = X.transpose()
    Y = Y.transpose()
    Z = Z.transpose()
    Zp = Z
    Zp = Zp - 0.05*Y
    w_1 = np.hamming(M)
    w_2 = np.hamming(N)
    W1,W2 = np.meshgrid(w_1,w_2)
147 W = W1*W2
```

```

fft = scipy.fftpack.fft2(Zp)
fftshift = scipy.fftpack.fftshift(fft)
S1 = abs(fftshift)**2
S1 = np.log10(S1)

Freq = range(-M/2,M/2)
FreqX = [k/float(0.008*M) for k in Freq]
Freq = range(-N/2,N/2)
157 FreqY = [k/float(0.008*N) for k in Freq]
xf, yf = np.meshgrid(FreqX,FreqY)
r = xf
for k in range(0,M):
    for j in range(0,N):
        r[j,k] = FreqX[k]*FreqX[k] + FreqY[j]*FreqY[j]
        r[j,k] = np.sqrt(r[j,k])

S1_1 = S1
r_1 = r
167 YS = np.reshape(S1_1,(M*N,1))
XR = np.reshape(r_1,(M*N,1))

index = []

for count in range(0,M*N):
    if XR[count] < 1e-6:
        if XR[count] == 0:
            f0 = YS[count]
            index.append(count)
177 print XR[count]
XR = np.delete(XR,index)
YS = np.delete(YS,index)
YS =YS
print XR.min()
XR = np.log10(XR)
nBIN =15
n, _ = np.histogram(XR, bins=nBIN)
sy, _ = np.histogram(XR, bins=nBIN, weights=YS)
sy2, _ = np.histogram(XR, bins=nBIN, weights=YS*YS)
187 mean = sy / n
std = np.sqrt(sy2/n - mean*mean)

print std
mean = 10**(mean)
std = 10**(std)
std = (std*std*mean -mean)/(1+std*std)
fmean = 10*((_[1:] + _[:-1])/2)

197 ax1.set_xscale("log", nonposx='clip')
ax1.set_yscale("log", nonposy='clip')
if legText == 'Experiment':
    ax1.errorbar(fmean, mean, yerr=std, fmt=color, label=legText)
else:
    ax1.plot(fmean, mean, marker = markerType, markevery=markevery[i],
            markeredgewidth= '0', color = color, label=legText, lw=0.5)
ax1.set_xlim([0.5, 1e2])
ax1.set_ylim([10, 1e12])
ax1.yaxis.set_ticks([10,1e4,1e8,1e12])
207 ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks_position('bottom')
ax1.get_yaxis().set_tick_params(which='minor', size=0)
ax1.get_yaxis().set_tick_params(which='minor', width=0)

if i == 1:
    legend = ax1.legend(loc='upper right', prop={'size':8})
    legend.get_frame().set_linewidth(0.0)
217 ax1.axes.get_xaxis().set_ticks([])
plt.ylabel(r'PSD ( $\mathbf{m}^4\mathbf{s}^{-1}$ )', fontsize=12)

```

Appendix D. Selected Python scripts: Data analysis and visualization

```
if i == 2:
    ax1.axes.get_xaxis().set_ticks([])
    ax1.axes.get_yaxis().set_ticks([])
elif i == 3:
    ax1.axes.get_xaxis().set_ticks([])
    plt.ylabel(r'PSD ( $\mathbf{m}^4$ )', fontsize=12)
elif i == 4:
    ax1.axes.get_xaxis().set_ticks([])
227 ax1.axes.get_yaxis().set_ticks([])
elif i == 5:
    plt.ylabel(r'PSD ( $\mathbf{m}^4$ )', fontsize=12)
    plt.xlabel(r'Wave number\, ( $\mathbf{m}^{-1}$ )', fontsize=12)
elif i == 6:
    plt.xlabel(r'Wave number\, ( $\mathbf{m}^{-1}$ )', fontsize=12)
    ax1.axes.get_yaxis().set_ticks([])
    r[119,60] = 0
for k in range(0,M):
    for j in range(0,N):
237         if r[j,k] > f_cr:
            fftshift[j,k] = 0

S2 = abs(fftshift)**2
f_ishift = np.fft.ifftshift(fftshift) #inverse shift
img_back = np.fft.ifft2(f_ishift) #inverse fourier transform
img_back = np.abs(img_back)
img_back = img_back + 0.05*Y
target = open('test.dat', 'w')

247 for j in range(0,N):
    for k in range(0,M):
        target.write(str(X[j,k]))
        target.write(' ')
        target.write(str(Y[j,k]))
        target.write(' ')
        target.write(str(abs(img_back[j,k])))
        target.write("\n")
    return

257 def processing_Phase(my_data,i,fileName,ax1,ax2,color,legText,markerType,markevery):
    #calculating the avsegae fft magnitude

    x = my_data[i,1:,0]
    y = my_data[i,1:,1]
    z = abs(my_data[i,1:,2])
    X = np.reshape(x,(M,N))
    Y = np.reshape(y,(M,N))
    Z = np.reshape(z,(M,N))
267 X = X.transpose()
    Y = Y.transpose()
    Z = Z.transpose()
    Zp = Z
    Zp = Zp - 0.05*Y
    w_1 = np.hamming(M)
    w_2 = np.hamming(N)
    W1,W2 = np.meshgrid(w_1,w_2)
    W = W1*W2

277 fft = scipy.fftpack.fft2(Zp)
    fftshift = scipy.fftpack.fftshift(fft)
    S1 = np.angle(fftshift)
    Freq = range(-M/2,M/2)
    FreqX = [k/float(0.008*M) for k in Freq]
    Freq = range(-N/2,N/2)
    FreqY = [k/float(0.008*N) for k in Freq]
    xf, yf = np.meshgrid(FreqX,FreqY)
    r = xf
    for k in range(0,M):
287         for j in range(0,N):
            r[j,k] = FreqX[k]*FreqX[k] + FreqY[j]*FreqY[j]
            r[j,k] = np.sqrt(r[j,k])
```

```

S1_1 = S1
r_1 = r
YS = np.reshape(S1_1,(M*N,1))
XR = np.reshape(r_1,(M*N,1))

index = []

297 for count in range(0,M*N):
    if XR[count] < 1e-6:
        if XR[count] == 0:
            f0 = YS[count]
            index.append(count)
            print XR[count]
XR = np.delete(XR,index)
YS = np.delete(YS,index)
YS =YS
print XR.min()
307 XR = np.log10(XR)

nBIN =15
n, _ = np.histogram(XR, bins=nBIN)
sy, _ = np.histogram(XR, bins=nBIN, weights=YS)
sy2, _ = np.histogram(XR, bins=nBIN, weights=YS*YS)
mean = sy / n
std = np.sqrt(sy2/n - mean*mean)
fmean = 10*((_[1:] + _[:-1])/2)
ax1.set_xscale("log", nonposx='clip')
317 if legText == 'Experiment':
    ax1.errorbar(fmean, mean, yerr=std, fmt=color, label=legText)
else:
    ax1.plot(fmean, mean, marker = markerType, markevery=markevery[i],
            markeredgewidth= '0', color = color, label=legText, lw=0.5)
ax1.set_xlim([0.5, 1e2])
ax1.set_ylim([-math.pi, math.pi])
ax1.yaxis.set_ticks([-math.pi,0,math.pi])
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
327 ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks_position('bottom')
ax1.get_yaxis().set_tick_params(which='minor', size=0)
ax1.get_yaxis().set_tick_params(which='minor', width=0)

if i == 1:
    legend = ax1.legend(loc='upper right', prop={'size':8})
    legend.get_frame().set_linewidth(0.0)
    ax1.axes.get_xaxis().set_ticks([])
    plt.ylabel(r'PSD ( $\mathbf{m}^4$ )', fontsize=12)
337 if i == 2:
    ax1.axes.get_xaxis().set_ticks([])
    ax1.axes.get_yaxis().set_ticks([])
    elif i == 3:
    ax1.axes.get_xaxis().set_ticks([])
    plt.ylabel(r'PSD ( $\mathbf{m}^4$ )', fontsize=12)
    elif i == 4:
    ax1.axes.get_xaxis().set_ticks([])
    ax1.axes.get_yaxis().set_ticks([])
    elif i == 5:
    plt.ylabel(r'PSD ( $\mathbf{m}^4$ )', fontsize=12)
    plt.xlabel(r'Wave number\, ( $\mathbf{m}^{-1}$ )', fontsize=12)
    elif i == 6:
    plt.xlabel(r'Wave number\, ( $\mathbf{m}^{-1}$ )', fontsize=12)
    ax1.axes.get_yaxis().set_ticks([])

#np.savetxt(fileName+'.dat',my_data[1,1:,0:3],delimiter=' ')
return
357

#-----Model Z-----

```

Appendix D. Selected Python scripts: Data analysis and visualization

```
def z_model_exp(my_data, my_datal):

    title = [r"0.25 h",r"0.5 h", r"1 h", r"2 h", r"4 h", r"8 h", r"16 h"]
    f3 = plt.figure(figsize= (12, 6))
    for i in range(1, 7):
        x = my_data[i,1:,0]/1000.0
        y = my_data[i,1:,1]/1000.0
        z = abs(my_data[i,1:,2])

        X = np.reshape(x,(M,N))
        Y = np.reshape(y,(M,N))
        Z = np.reshape(z,(M,N))

        levels = np.arange(0,170,5)
        X = X.transpose()
        Y = Y.transpose()
        Z = Z.transpose()

        extent = (min(x), max(x),min(y), max(y))

        levels = np.arange(0,180,10)
        ax1 =plt.subplot(2,6,i,aspect='equal')
        ax1.tick_params(axis='both', which='major', labelsize=8)
        CS1 = ax1.contourf(X,Y,Z, levels, vmax=180, vmin=0, cmap='rainbow')
        ax1.axis([x.min(), x.max(), y.min(), y.max()])
        plt.title(title[i])
        if i ==1:
            ax1.axes.get_xaxis().set_ticks([])
            plt.ylabel("$y$ (m)")
            ax1.text(-1.1, 1.0, 'Experiment',horizontalalignment='center',
                    verticalalignment='center', rotation = 90, fontsize=12)
        if i ==2:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==3:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==4:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==5:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==6:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        f3.subplots_adjust(right=0.85)
        cbar_ax = f3.add_axes([0.86, 0.55, 0.01, 0.3])
        cbar1 = plt.colorbar(CS1,cbar_ax, ticks=np.arange(0,180,20),extend='neither',
                             spacing=0.0002, orientation='vertical')
        cbar1.set_label("$z$ (mm)")
    for i in range(1, 7):
        x = my_datal[i,1:,0]/1000.0
        y = my_datal[i,1:,1]/1000.0
        z = abs(my_datal[i,1:,2])
        X = np.reshape(x,(M,N))
        Y = np.reshape(y,(M,N))
        Z = np.reshape(z,(M,N))
        X = X.transpose()
        Y = Y.transpose()
        Z = Z.transpose()

        extent = (min(x), max(x),min(y), max(y))
        levels = np.arange(0,180,10)
        ax2 =plt.subplot(2,6,i+6,aspect='equal')
        ax2.tick_params(axis='both', which='major', labelsize=8)
        CS2 = ax2.contourf(X,Y,Z, levels, vmax=180, vmin=0,cmap='rainbow')
        ax2.axis([x.min(), x.max(), y.min(), y.max()])
        if i ==1:
            plt.xlabel("$x$ (m)")
```

```

plt.ylabel("$y$ (m)")
ax2.text(-1.1, 1.0, 'Model',horizontalalignment='center',
verticalalignment='center', rotation = 90, fontsize=12)
if i ==2:
plt.xlabel("$x$ (m)")
437 ax1.axes.get_yaxis().set_ticks([])

if i ==3:
plt.xlabel("$x$ (m)")
ax1.axes.get_yaxis().set_ticks([])

if i ==4:
plt.xlabel("$x$ (m)")
ax1.axes.get_yaxis().set_ticks([])
447 if i ==5:
plt.xlabel("$x$ (m)")
ax1.axes.get_yaxis().set_ticks([])

if i ==6:
plt.xlabel("$x$ (m)")
ax1.axes.get_yaxis().set_ticks([])

f3.savefig(outFile+'Zmodelxp'+'.png',papertype = 'a4', bbox_inches='tight', dpi = 500)
#-----PSD contours models 1-----
def contour_PSD_model():
457 title = [r"0.25 h",r"0.5 h", r"1 h", r"2 h", r"4 h", r"8 h", r"16 h"]
f2 = plt.figure(figsize= (10, 4))
Freq = range(-M/2+1,M/2+1)
FreqX = [k/float(0.008*M) for k in Freq]
Freq = range(-N/2+1,N/2+1)
FreqY = [k/float(0.008*N) for k in Freq]

for i in range(1, 7):
x = my_data[i,1:,0]
y = my_data[i,1:,1]
467 z = abs(my_data[i,1:,2])
X = np.reshape(x,(M,N))
Y = np.reshape(y,(M,N))
Z = np.reshape(z,(M,N))
X = X.transpose()
Y = Y.transpose()
Z = Z.transpose()
Zp = Z
fft = scipy.fftpack.fft2(Zp)
fftshift = scipy.fftpack.fftshift(fft)
477 S1 = abs(fftshift)**2+1
#S1 = np.log10(S1)
xf, yf = np.meshgrid(FreqX,FreqY)
ax1 =plt.subplot(2,6,i,aspect='equal')
ax1.tick_params(axis='both', which='major', labelsize=8)
CS =plt.pcolor(xf, yf, S1, norm=LogNorm(vmin = 1, vmax=1e7))
plt.title(title[i])

if i ==1:
ax1.axes.get_xaxis().set_ticks([])
487 plt.ylabel(r"$w_{y}$\,\,${\mathregular{\rm m}^{-1}}$")
ax1.text(-150, 0, 'Experiment',horizontalalignment='center',
verticalalignment='center',color = 'r', fontsize=12,rotation=90)
if i ==2:
ax1.axes.get_xaxis().set_ticks([])
ax1.axes.get_yaxis().set_ticks([])

if i ==3:
ax1.axes.get_xaxis().set_ticks([])
ax1.axes.get_yaxis().set_ticks([])
497 if i ==4:
ax1.axes.get_xaxis().set_ticks([])
ax1.axes.get_yaxis().set_ticks([])
if i ==5:
ax1.axes.get_xaxis().set_ticks([])

```

Appendix D. Selected Python scripts: Data analysis and visualization

```
ax1.axes.get_yaxis().set_ticks([])

if i ==6:
    ax1.axes.get_xaxis().set_ticks([])
    ax1.axes.get_yaxis().set_ticks([])

f2.subplots_adjust(right=0.92)
plt.axis([xf.min(), xf.max(), yf.min(), yf.max()])
cbar_ax = f2.add_axes([0.93, 0.2, 0.02, 0.5])
cbar1 = plt.colorbar(CS,cbar_ax,extend='neither', spacing=0.0002,
orientation='vertical',ticks=[1, 10, 100,1e3,1e4,1e5,1e6,1e7])
cbar1.set_label("PSD\,$(\mathregular{\mathrm{4}})$")
cbar1.ax.minorticks_off()

517 ## model:
for i in range(1, 7):
    x = my_data1[i,1:,0]
    y = my_data1[i,1:,1]
    z = abs(my_data1[i,1:,2])
    X = np.reshape(x,(M,N))
    Y = np.reshape(y,(M,N))
    Z = np.reshape(z,(M,N))
    X = X.transpose()
    Y = Y.transpose()
    Z = Z.transpose()
    Zp = Z
    fft = scipy.fftpack.fft2(Zp)
    fftshift = scipy.fftpack.fftshift(fft)
    S1 = abs(fftshift)**2 + 1
    #S1 = np.log10(S1)
    xf, yf = np.meshgrid(FreqX,FreqY)
    ax1=plt.subplot(2,6,i+6,aspect='equal')
    ax1.tick_params(axis='both', which='major', labelsize=8)
    #plt.title(title[i])
    CS=plt.pcolor(xf, yf, S1, norm=LogNorm(vmin = 1, vmax=1e7))
    plt.axis([xf.min(), xf.max(), yf.min(), yf.max()])
    if i ==1:
        plt.xlabel(r"$w_x\backslash,(\mathregular{\mathrm{-1}})$")
        plt.ylabel(r"$w_y\backslash,(\mathregular{\mathrm{-1}})$")
        ax1.text(-150, 0, 'Model',horizontalalignment='center',
verticalalignment='center',color = 'r', fontsize=12,rotation=90)
    if i ==2:
        plt.xlabel(r"$w_x\backslash,(\mathregular{\mathrm{-1}})$")
        ax1.axes.get_yaxis().set_ticks([])
    if i ==3:
        plt.xlabel(r"$w_x\backslash,(\mathregular{\mathrm{-1}})$")
        ax1.axes.get_yaxis().set_ticks([])
    if i ==4:
        plt.xlabel(r"$w_x\backslash,(\mathregular{\mathrm{-1}})$")
        ax1.axes.get_yaxis().set_ticks([])
    if i ==5:
        plt.xlabel(r"$w_x\backslash,(\mathregular{\mathrm{-1}})$")
        ax1.axes.get_yaxis().set_ticks([])
    if i ==6:
        plt.xlabel(r"$w_x\backslash,(\mathregular{\mathrm{-1}})$")
        ax1.axes.get_yaxis().set_ticks([])
    f2.savefig(outFile+'SpectralContours_Model_Exp'+'.png', bbox_inches='tight', dpi = 500)
--
##-----PSD avergae in each bin-----
def ave_PSD():
    left, width = 1e-5, 1e2
    bottom, height = 1e1, 1e12
    right = left + width
    top = bottom + height

    legText = ["Experiment", "Model"]
    color = ['k','g','b']
    markerType = ['o','o', 'v', '^', '<', '>', 's']
    title = ["(a) 0.25\,$h$, "(b) 0.5\,$h$, "(c) 1\,$h$, "(d) 2\,$h$, "(e) 4\,$h$, "(f) 8\,$h$, "(g) 16\,$h$"]
```

```

markevery = [1,1,1,1,1,1,1]
f1 = plt.figure(figsize= (8, 7))
ax2 = 1
577
for i in range(1,7):
    ax1 =plt.subplot(3,2,i,aspect=0.11)
    ax1.text(1e-1*right, 0.95*top, title[i],horizontalalignment='right', verticalalignment='top',fontsize=12)
    ax1.tick_params(axis='both', which='major', labels=10)
    processing(my_data,i,fileName0[i],ax1,ax2,color[0], legText[0],markerType[1],markevery)
    processing(my_data1,i,fileName1[i],ax1,ax2,color[1], legText[1],markerType[2],markevery)
f1.savefig(outFile+'E_K'+'.png',papertype= 'a4', bbox_inches='tight', dpi = 500)

587 #-----Model Q-----

def discharge_model_exp(my_data, my_data1):

    #print(my_data)
    title = ["(a) 0.25$, $h", "(b) 0.5$, $h", "(c) 1$, $h", "(d) 2$, $h", "(e) 4$, $h", "(f) 8$, $h", "(g) 16$, $h"]
    f2 = plt.figure(figsize= (10, 5))
    for i in range(1, 7):
        x = my_data[i,1:,0]/1000.0
        y = my_data[i,1:,1]/1000.0
597 q = my_data[i,1:,5]
        X = np.reshape(x,(M,N))
        Y = np.reshape(y,(M,N))
        Q = np.reshape(q,(M,N))
        X = X.transpose()
        Y = Y.transpose()
        Q = Q.transpose()
        Q = Q + 1
        extent = (min(x), max(x),min(y), max(y))
        ax1 =plt.subplot(2,6,i,aspect='equal')
        ax1.tick_params(axis='both', which='major', labels=8)
607 plt.title(title[i])
        levels = np.arange(1, 8, 0.1)
        cmap = plt.get_cmap('rainbow')#rainbow coolwarm Blues
        norm = BoundaryNorm(levels, ncolors=cmap.N, clip=False)
        CS1 =plt.pcolor(X, Y, Q, norm=LogNorm(vmin = 10, vmax=Q.max()), cmap='rainbow')
        ax1.axis([x.min(), x.max(), y.min(), y.max()])
        if i ==1:
            ax1.axes.get_xaxis().set_ticks([])
            plt.ylabel("$y$ (m)")
617 ax1.text(-1.2, 1.0, 'Experiment',horizontalalignment='center', verticalalignment='center', rotation = 90, fontsize=12)
        if i ==2:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==3:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==4:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
627 if i ==5:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        if i ==6:
            ax1.axes.get_xaxis().set_ticks([])
            ax1.axes.get_yaxis().set_ticks([])
        f2.subplots_adjust(right=0.85)

        cbar_ax = f2.add_axes([0.86, 0.55, 0.01, 0.3])
        cbar1 = plt.colorbar(CS1,cbar_ax,extend='neither', spacing=0.0002,
637 orientation='vertical',ticks=[1, 10, 100,1e3,1e4,1e5,1e6,1e7,1e8])
        cbar1.set_label("$Q$, $({\rm mm}^3\,{\rm h}^{-1})$")
    for i in range(1, 7):
        x = my_data1[i,1:,0]/1000.0
        y = my_data1[i,1:,1]/1000.0
        q = my_data1[i,1:,5]
        X = np.reshape(x,(M,N))
        Y = np.reshape(y,(M,N))

```

Appendix D. Selected Python scripts: Data analysis and visualization

```
Q = np.reshape(q, (M,N))
X = X.transpose()
647 Y = Y.transpose()
Q = Q.transpose()
Q = Q + 1
extent = (min(x), max(x), min(y), max(y))

ax1 = plt.subplot(2,6,i+6, aspect='equal')
ax1.tick_params(axis='both', which='major', labelsize=8)
levels = np.arange(1, 8, 0.1)
cmap = plt.get_cmap('rainbow') #rainbow coolwarm Blues
norm = BoundaryNorm(levels, ncolors=cmap.N, clip=False)
657 CSI = plt.pcolor(X, Y, Q, norm=LogNorm(vmin = 10, vmax=Q.max()), cmap='rainbow')
ax1.axis([x.min(), x.max(), y.min(), y.max()])
if i == 1:
    plt.xlabel("$x$ (m)")
    plt.ylabel("$y$ (m)")
    ax1.text(-1.2, 1.0, 'Model', horizontalalignment='center',
            verticalalignment='center', rotation = 90, fontsize=12)
    plt.xticks([-0.4, 0, 0.4])
if i == 2:
    plt.xlabel("$x$ (m)")
667 ax1.axes.get_yaxis().set_ticks([])
    plt.xticks([-0.4, 0, 0.4])
if i == 3:
    plt.xlabel("$x$ (m)")
    ax1.axes.get_yaxis().set_ticks([])
    plt.xticks([-0.4, 0, 0.4])
if i == 4:
    plt.xlabel("$x$ (m)")
    ax1.axes.get_yaxis().set_ticks([])
    plt.xticks([-0.4, 0, 0.4])
677 if i == 5:
    plt.xlabel("$x$ (m)")
    ax1.axes.get_yaxis().set_ticks([])
    plt.xticks([-0.4, 0, 0.4])
if i == 6:
    plt.xlabel("$x$ (m)")
    ax1.axes.get_yaxis().set_ticks([])
    plt.xticks([-0.4, 0, 0.4])

f2.savefig(outFile+'Qmodellexp'+'.png', papertype = 'a4', bbox_inches='tight', dpi = 500)
687 #-----Exceedance Probability of Discharge-----
def exceedance_Q(my_data, my_data1, my_data2):

    title = ["(a) 0.25$\\$, $h", "(b) 0.5$\\$, $h", "(c) 1$\\$, $h", "(d) 2$\\$, $h", "(e) 4$\\$, $h", "(f) 8$\\$, $h", "(g) 16$\\$, $h"]
    f8 = plt.figure(figsize= (6, 6))
    for i in range(1, 7):
        ax1 = plt.subplot(3,2,i)
        q = my_data[i,1:,5]
        qselect, ExProbq = Exceedance(q)
697 ax1.loglog(qselect, ExProbq, linewidth=1.5, color='black', label='Experiment')

        left, width = 1e0, 1e9
        bottom, height = 1e-5, 5e0+1e-5
        right = left + width
        top = bottom + height
        q = my_data1[i,1:,5]
        qselect, ExProbq = Exceedance(q)
        markers_on = [20, 40, 60, 80]
707 ax1.loglog(qselect, ExProbq, linestyle='--', color='red', marker='o', linewidth=1.5,
            markevery=10, markeredgewidth=0, label='Exp. (filtered)')

        markers_on = [25, 45, 65, 85]
        q = my_data2[i,1:,5]
        qselect, ExProbq = Exceedance(q)
        ax1.loglog(qselect, ExProbq, linestyle='--', marker='v', color='green', linewidth=1.5,
            markevery=15, markeredgewidth=0, label='Model')
        ax1.set_ylim([1e-5, 2e0])
        start, end = ax1.get_ylim()
```



```

axl.yaxis.set_ticks([1e-4, 1e-2, 1e0])
717 axl.text(0.95*right, 0.95*top, title[i],horizontalalignment='right', verticalalignment='top',fontsize=12)
axl.spines['right'].set_visible(False)
axl.spines['top'].set_visible(False)
axl.yaxis.set_ticks_position('left')
axl.xaxis.set_ticks_position('bottom')

if i == 1:
    plt.ylabel("$P$", $Q$>$q$)", fontsize=12)
    legend = axl.legend(loc='lower left', prop={'size':8})
    legend.get_frame().set_linewidth(0.0)
727 axl.axes.get_xaxis().set_ticks([])
if i == 2:
    axl.axes.get_xaxis().set_ticks([])
    axl.axes.get_yaxis().set_ticks([])
if i == 3:
    plt.ylabel("$P$", $Q$>$q$)", fontsize=12)
    axl.axes.get_xaxis().set_ticks([])
if i == 4:
    axl.axes.get_xaxis().set_ticks([])
    axl.axes.get_yaxis().set_ticks([])
737 if i == 5:
    plt.ylabel("$P$", $Q$>$q$)", fontsize=12)
    plt.xlabel("$q$", $Q$>$q$)", fontsize=12)
    axl.xaxis.set_ticks([1e0, 1e2, 1e4, 1e6, 1e8])
if i == 6:
    plt.xlabel("$q$", $Q$>$q$)", fontsize=12)
    axl.axes.get_yaxis().set_ticks([])
    axl.xaxis.set_ticks([1e0, 1e2, 1e4, 1e6, 1e8])
    axl.set_xlim([1e0, 1e9])
    start, end = axl.get_xlim()
747 f8.savefig(outFile+'PDF_QmodelExp'+'.png',papertype = 'a4', bbox_inches='tight', dpi = 500)

#-----Hack's low-----

def Hacks():
    title = ["(a) 0.25$\\$,Sh", "(b) 0.5$\\$,Sh", "(c) 1$\\$,Sh", "(d) 2$\\$,Sh", "(e) 4$\\$,Sh", "(f) 8$\\$,Sh", "(g) 16$\\$,Sh"]

    titlesize=14
    axistitle = 12
757 ticksize = 10
    left, width = 20, 2*1e6
    bottom, height = 5, 1e4
    right = left + width
    top = bottom + height
    f1 = plt.figure(figsize= (6,6))
    for i in range(1, 7):
        x = my_data[i,1:,6]
        y = my_data[i,1:,7]
        xlog = np.log10(x)
        ylog = np.log10(y)
767 axl=plt.subplot(3,2,i)
        axl.text(0.4*right, 0.95*top, title[i],horizontalalignment='right', verticalalignment='top',fontsize=12)
        axl.tick_params(axis='both', which='major', labelsize=ticksize)
        nBIN =20
        n, _ = np.histogram(xlog, bins=nBIN)
        sy, _ = np.histogram(xlog, bins=nBIN, weights=ylog)
        meanl = sy / n
        xmeanlog = (_[1:] + _[:-1])/2
        xmean = 10**xmeanlog
777 axl.scatter(xmean,10**meanl,marker = 's',color='k',facecolors='none',label='Experiment')
        a = [0,0,0,0]
        x1 = np.linspace(np.min(xmeanlog), np.max(xmeanlog), 6)
        a[0], b = linearPartFit(xmean,10**meanl,200,1e4)
        y1 = b+a[0]*x1
        axl.plot(10**x1,10**y1,color='k')
    for i in range(1, 7):
        x = my_data[i,1:,6]
        y = my_data[i,1:,7]
        xlog = np.log10(x)

```

Appendix D. Selected Python scripts: Data analysis and visualization

```
787 ylog = np.log10(y)
    ax1 = plt.subplot(3,2,i)
    ax1.tick_params(axis='both', which='major', labelsize=ticksiz)
    nBIN =20
    n, _ = np.histogram(xlog, bins=nBIN)
    sy, _ = np.histogram(xlog, bins=nBIN, weights=ylog)
    mean1 = sy / n
    xmeanlog = (_[1:] + _[:-1])/2
    xmean = 10**xmeanlog
    ax1.scatter(xmean,10**mean1,marker = 'o',color='r',facecolors='none',label='Exp. (filtered)')

797 for i in range(1, 7):
    x = my_data2[i,1:,6]
    y = my_data2[i,1:,7]
    xlog = np.log10(x)
    ylog = np.log10(y)
    ax1 = plt.subplot(3,2,i)
    ax1.tick_params(axis='both', which='major', labelsize=ticksiz)
    nBIN =20
    n, _ = np.histogram(xlog, bins=nBIN)
807 sy, _ = np.histogram(xlog, bins=nBIN, weights=ylog)
    mean1 = sy / n
    xmeanlog = (_[1:] + _[:-1])/2
    xmean = 10**xmeanlog
    ax1.scatter(xmean,10**mean1,marker = 'v',color='g',facecolors='none',label='Model')
    ax1.set_yscale('log')
    ax1.set_xscale('log')
    ax1.axis([x.min(), x.max(), y.min(), 10**1])
    ax1.spines['right'].set_visible(False)
    ax1.spines['top'].set_visible(False)
817 ax1.yaxis.set_ticks_position('left')
    ax1.xaxis.set_ticks_position('bottom')
    ax1.set_ylim([5, 1e4])
    ax1.yaxis.labelpad = 20
    if i == 1:
        plt.ylabel(r"$I\,,$(mm)",rotation = 'horizontal',fontsize = axistitle)
        legend = ax1.legend(loc='lower right', prop={'size':8})
        legend.get_frame().set_linewidth(0.0)
        ax1.axes.get_xaxis().set_ticks([])
    if i == 2:
827 ax1.axes.get_xaxis().set_ticks([])
        ax1.axes.get_yaxis().set_ticks([])
    if i == 3:
        plt.ylabel(r"$I\,,$(mm)",rotation = 'horizontal',fontsize = axistitle)
        ax1.axes.get_xaxis().set_ticks([])
    if i == 4:
        ax1.axes.get_xaxis().set_ticks([])
        ax1.axes.get_yaxis().set_ticks([])
    if i == 5:
        plt.ylabel(r"$I\,,$(mm)",rotation = 'horizontal',fontsize = axistitle)
837 plt.xlabel("$A\,,\mathregular{(mm^2)}$",fontsize = axistitle)
    if i == 6:
        plt.xlabel("$A\,,\mathregular{(mm^2)}$",fontsize = axistitle)
        ax1.axes.get_yaxis().set_ticks([])
    plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.35, wspace=0.35)
    f1.savefig(outFile+'HACK_exp_Binn'+'.png',papertype = 'a4', bbox_inches='tight', dpi = 500)

z_model_exp(my_data, my_data1)
discharge_model_exp(my_data, my_data1)
contour_PSD_model()
847 ave_PSD()
Hacks()
exceedance_Q(my_data,my_data1,my_data2)
```

Bibliography

- Abbaspour, K., M. Faramarzi, S. Ghasemi, and Y. Yang, Assessing the Impact of Climate Change on Water Resources in Iran, *Water Resources Research*, 45(10), doi:10.1029/2008WR007615, 2009.
- Abed-Elmdoust, A., M.-A. Miri, and A. Singh, Reorganization of river networks under changing spatiotemporal precipitation patterns: An optimal channel network approach, *Water Resources Research*, 52(11), 8845–8860, doi:10.1002/2015WR018391, 2016.
- Ahnert, F., Brief Description of a Comprehensive Three-Dimensional Process-Response Model of Landform Development, *Zeitschrift für Geomorphologie Supplemental*, 25, 29–49, 1976.
- Ahnert, F., Some Comments on the Quantitative Formulation of Geomorphological Processes in a Theoretical Model, *Earth Surface Processes*, 2(2-3), 191–201, doi:10.1002/esp.3290020211, 1977.
- Ahnert, F., Approaches to Dynamic Equilibrium in Theoretical Simulations of Slope Development, *Earth Surface Processes and Landforms*, 12(1), 3–15, doi:10.1002/esp.3290120103, 1987.
- Aich, V., A. Zimmermann, and H. Elsenbeer, Quantification and Interpretation of Suspended-Sediment Discharge Hysteresis Patterns: How Much Data Do We Need?, *Catena*, 122, 120–129, doi:10.1016/j.catena.2014.06.020, 2014.
- Alemayehu, W., E. Teklu, and D. Prhaba, Sediment and Nutrient Lost by Runoff from Two Watersheds, Digga District in Blue Nile Basin, Ethiopia, *African Journal of Environmental Science and Technology*, 8(9), 498–510, doi:10.5897/AJEST2014.1747, 2014.

Bibliography

- Ancey, C., P. Bohorquez, and J. Heyman, Stochastic Interpretation of the Advection-Diffusion Equation and Its Relevance to Bed Load Transport, *Journal of Geophysical Research: Earth Surface*, 120(12), 2529–2551, doi:10.1002/2014JF003421, 2015.
- Arjmand Sajjadi, S., and M. Mahmoodabadi, Sediment Concentration and Hydraulic Characteristics of Rain-Induced Overland Flows in Arid Land Soils, *Journal of Soils and Sediments*, 15(3), 710–721, doi:10.1007/s11368-015-1072-z, 2015.
- Armstrong, A., J. N. Quinton, B. C. P. Heng, and G. C. Sander, Processes Controlling the Development of a Shielding Layer on Natural Soil, *European journal of soil science*, 63(1), 54–64, doi:https://doi.org/10.1111/j.1365-2389.2011.01416.x, 2012.
- Azari, M., H. R. Moradi, B. Saghaian, and M. Faramarzi, Climate Change Impacts on Streamflow and Sediment Yield in the North of Iran, *Hydrological Sciences Journal*, 61(1), 123–133, doi:10.1080/02626667.2014.967695, 2016.
- Babault, J., S. Bonnet, A. Crave, and J. Van Den Driessche, Influence of Piedmont Sedimentation on Erosion Dynamics of an Uplifting Landscape: An Experimental Approach, *Geology*, 33 (4), 301–304, doi:10.1130/G21095.1, 2005.
- Babault, J., S. Bonnet, J. V. D. Driessche, and A. Crave, High Elevation of Low-Relief Surfaces in Mountain Belts: Does It Equate to Post-Orogenic Surface Uplift?, *Terra Nova*, 19(4), 272–277, doi:10.1111/j.1365-3121.2007.00746.x, 2007.
- Bak, P., C. Tang, and K. Wiesenfeld, Self-Organized Criticality, *Physical Review A*, 38(1), 364–374, doi:10.1103/PhysRevA.38.364, 1988.
- Ballato, P., A. Landgraf, T. F. Schildgen, D. F. Stockli, M. Fox, M. R. Ghassemi, E. Kirby, and M. R. Strecker, The Growth of a Mountain Belt Forced by Base-Level Fall: Tectonics and Surface Processes during the Evolution of the Alborz Mountains, N Iran, *Earth and Planetary Science Letters*, 425, 204–218, doi:10.1016/j.epsl.2015.05.051, 2015.
- Banavar, J. R., A. Maritan, and A. Rinaldo, Size and Form in Efficient Transportation Networks, *Nature*, 399, 130–132, doi:10.1038/20144, 1999.
- Banavar, J. R., F. Colaiori, A. Flammini, A. Maritan, and A. Rinaldo, Scaling, Optimality, and Landscape Evolution, *Journal of Statistical Physics*, 104(1-2), 1–48, doi:10.1023/A:1010397325029, 2001.

- Baril, P., Erodibilité Des Sols et Érodabilité Des Terres : Application Au Plateau Vaudois, Ph.D. Thesis, Ecole Polytech. Féd. de Lausanne (EPFL), doi:10.5075/epfl-thesis-940, 1991.
- Barry, D., G. Sander, S. Jomaa, B. Heng, J.-Y. Parlange, I. Lisle, and W. Hogarth, Exact Solutions of the Hairsine–Rose Precipitation-Driven Erosion Model for a Uniform Grain-Sized Soil, *Journal of Hydrology*, 389(3-4), 399–405, doi:10.1016/j.jhydrol.2010.06.016, 2010.
- Batalla, R. J., C. M. Gómez, and G. M. Kondolf, Reservoir-Induced Hydrological Changes in the Ebro River Basin (NE Spain), *Journal of Hydrology*, 290(1-2), 117–136, doi:10.1016/j.jhydrol.2003.12.002, 2004.
- Baumgarten, A. S., and K. Kamrin, A General Fluid-Sediment Mixture Model and Constitutive Theory, 2018.
- Benaïchouche, A., O. Stab, B. Tessier, and I. Cojan, Evaluation of a landscape evolution model to simulate stream piracy: Insights from multivariable numerical tests using the example of the Meuse basin, France, *Geomorphology*, 253, 168–180, doi:10.1016/j.geomorph.2015.10.001, 2016.
- Bennett, S. J., and R. Liu, Basin Self-Similarity, Hack’s Law, and the Evolution of Experimental Rill Networks, *Geology*, 44(1), 35–38, doi:10.1130/G37214.1, 2015.
- Bennett, S. J., L. M. Gordon, V. Neroni, and R. R. Wells, Emergence, Persistence, and Organization of Rill Networks on a Soil-Mantled Experimental Landscape, *Natural Hazards*, 79(S1), 7–24, doi:10.1007/s11069-015-1599-8, 2015.
- Benson, D. A., M. M. Meerschaert, and J. Revielle, Fractional Calculus in Hydrologic Modeling: A Numerical Perspective, *Advances in Water Resources*, 51, 479–497, doi:10.1016/j.advwatres.2012.04.005, 2013.
- Berger, C., M. Schulze, D. Rieke-Zapp, and F. Schlunegger, Rill Development and Soil Erosion: A Laboratory Study of Slope and Rainfall Intensity, *Earth Surface Processes and Landforms*, 35(12), 1456–1467, doi:10.1002/esp.1989, 2010.
- Beuselinck, L., G. Govers, P. B. Hairsine, G. C. Sander, and M. Breynaert, The Influence of Rainfall on Sediment Transport by Overland Flow over Areas of Net Deposition, *Journal of Hydrology*, 257(1-4), 145–163, doi:10.1016/S0022-1694(01)00548-0, 2002.

Bibliography

- Bigi, A., L. E. Hasbargen, A. Montanari, and C. Paola, Knickpoints and Hillslope Failures: Interactions in a Steady-State Experimental Landscape, *Geological Society of America Special Papers*, 398(March), 295–307, doi:10.1130/2006.2398(18), 2006.
- Birnir, B., Turbulent Rivers, *Quarterly of Applied Mathematics*, 66(3), 565–594, doi:https://doi.org/10.1090/S0033-569X-08-01123-8, 2008.
- Birnir, B., T. R. Smith, and G. E. Merchant, The Scaling of Fluvial Landscapes, *Computers and Geosciences*, 27(10), 1189–1216, doi:10.1016/S0098-3004(01)00022-X, 2001.
- Birnir, B., J. Hernandez, and T. R. Smith, The Stochastic Theory of Fluvial Landsurfaces, *Journal of Nonlinear Science*, 17(1), 13–57, doi:10.1007/s00332-005-0688-3, 2007.
- Blackman, R. B., and J. W. Tukey, The measurement of power spectra from the point of view of communications engineering - Part I, *Bell System Technical Journal*, 37(1), 185–282, doi:10.1002/j.1538-7305.1958.tb03874.x, 1958.
- Blanco-Canqui, H., and R. Lal, *Principles of Soil Conservation and Management*, Springer, Dordrecht, doi:10.1007/978-1-4020-8709-7, 2010.
- Bonetti, S., and A. Porporato, On the Dynamic Smoothing of Mountains, *Geophysical Research Letters*, 44(11), 5531–5539, doi:10.1002/2017GL073095, 2017.
- Bonnet, S., Shrinking and Splitting of Drainage Basins in Orogenic Landscapes from the Migration of the Main Drainage Divide, *Nature Geoscience*, 2(12), 897, doi:10.1038/ngeo700, 2009.
- Bonnet, S., and A. Crave, Landscape Response to Climate Change: Insights from Experimental Modeling and Implications for Tectonic versus Climatic Uplift of Topography, *Geology*, 31(2), 123–126, doi:10.1130/0091-7613(2003)031<0123:LRTCCI>2.0.CO;2, 2003.
- Bonnet, S., and A. Crave, Macroscale Dynamics of Experimental Landscapes, *Geological Society, London, Special Publications*, 253(1), 327–339, doi:10.1144/GSL.SP.2006.253.01.17, 2006.
- Brunton, D. A., and R. B. Bryan, Rill Network Development and Sediment Budgets, *Earth Surface Processes and Landforms*, 25(7), 783–800, doi:https://doi.org/10.1002/1096-9837(200007)25:7<783::AID-ESP106>3.0.CO;2-W, 2000.

- Bryan, R. B., and J. Poesen, Laboratory Experiments on the Influence of Slope Length on Runoff, Percolation and Rill Development, *Earth Surface Processes and Landforms*, 14, 211–231, doi:10.1002/esp.3290140304, 1989.
- Buendia, C., D. Vericat, R. J. Batalla, and C. N. Gibbins, Temporal Dynamics of Sediment Transport and Transient In-Channel Storage in a Highly Erodible Catchment, *Land Degradation & Development*, 27(4), 1045–1063, doi:10.1002/ldr.2348, 2016.
- Bussi, G., F. Francés, J. J. Montoya, and P. Y. Julien, Distributed Sediment Yield Modelling: Importance of Initial Sediment Conditions, *Environmental Modelling & Software*, 58, 58–70, doi:10.1016/J.ENVSOFT.2014.04.010, 2014.
- Caldarelli, G., A. Giacometti, A. Maritan, I. Rodriguez-Iturbe, and A. Rinaldo, Randomly Pinned Landscape Evolution, *Physical Review E*, 55(5), R4865–R4868, doi:10.1103/PhysRevE.55.R4865, 1997.
- Calvert, J., M. Balázs, and K. Michaelides, Unifying Particle-Based and Continuum Models of Hillslope Evolution with a Probabilistic Scaling Technique, 2018.
- Carson, M. A., and M. J. Kirkby, Hillslope Form and Process, *Science*, 178(4065), 1083–1084, doi:10.1126/science.178.4065.1083-a, 1972.
- Cauchy, A., Sur les résultats moyens d'observations de même nature, et sur les résultats les plus probables, *Comptes Rendus de l'Académie des Sciences*, 37, 198–206, 1853.
- Cerdà, A., Effects of Rock Fragment Cover on Soil Infiltration, Interrill Runoff and Erosion, *European Journal of Soil Science*, 52(1), 59–68, doi:10.1046/j.1365-2389.2001.00354.x, 2001.
- Cerdà, A., A. G. Morera, and M. B. Bodí, Soil and Water Losses from New Citrus Orchards Growing on Sloped Soils in the Western Mediterranean Basin, *Earth Surface Processes and Landforms*, 34(13), 1822–1830, doi:10.1002/esp.1889, 2009.
- Cerdà, A., J. Rodrigo-Comino, A. Novara, E. C. Brevik, A. R. Vaezi, M. Pulido, A. Giménez-Morera, and S. D. Keesstra, Long-Term Impact of Rainfed Agricultural Land Abandonment on Soil Erosion in the Western Mediterranean Basin , Long-Term Impact of Rainfed Agricultural Land Abandonment on Soil Erosion in the Western Mediterranean Basin, *Progress in Physical Geography: Earth and Environment*, 42(2), 202–219, doi:10.1177/0309133318758521, 2018.

Bibliography

- Certini, G., Effects of Fire on Properties of Forest Soils: A Review, *Oecologia*, 143(1), 1–10, doi:10.1007/s00442-004-1788-8, 2005.
- Chen, A., J. Darbon, G. Buttazzo, F. Santambrogio, and J. M. Morel, On the equations of landscape formation, *Interfaces and Free Boundaries*, 16(1), 105–136, doi:10.4171/IFB/315, 2014a.
- Chen, A., J. Darbon, and J.-M. Morel, Landscape Evolution Models: A Review of Their Fundamental Equations, *Geomorphology*, 219, 68–86, doi:10.1016/j.geomorph.2014.04.037, 2014b.
- Chen, F., Coupled Flow Discrete Element Method Application in Granular Porous Media Using Open Source Codes, Ph.D. Thesis, University of Tennessee, 2009.
- Christiansen, J. E., Irrigation by Sprinkling, California Agricultural Experiment Station Bulletin 670, Berkeley, <https://archive.org/stream/irrigationbyspri670chri>, 1942.
- Colby, B., Fluvial Sediments a Summary of Source, Transportation, Deposition, and Measurement of Sediment Discharge, *Report 1181A*, doi:10.3133/b1181A, 1963.
- Corenblit, D., N. S. Davies, J. Steiger, M. R. Gibling, and G. Bornette, Considering river structure and stability in the light of evolution: Feedbacks between riparian vegetation and hydrogeomorphology, *Earth Surface Processes and Landforms*, 40(2), 189–207, doi:10.1002/esp.3643, 2015.
- Costa-Cabral, M. C., and S. J. Burges, Digital Elevation Model Networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas, *Water Resources Research*, 30(6), 1681–1692, doi:10.1029/93WR03512, 1994.
- Crave, A., and P. Davy, Scaling Relationships of Channel Networks at Large Scales: Examples from Two Large-Magnitude Watersheds in Brittany, France, *Tectonophysics*, 269(1-2), 91–111, doi:10.1016/S0040-1951(96)00142-4, 1997.
- Crave, A., D. Lague, P. Davy, J. Kermarrec, D. Sokoutis, L. Bodet, and R. Compagnon, Analogue Modelling of Relief Dynamics, *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, 25(6), 549–553, doi:10.1016/S1464-1895(00)00084-3, 2000.
- Culling, W., Analytical theory of erosion, *The Journal of Geology*, 68(3), 336–344, doi:<https://doi.org/10.1086/626663>, 1960.

- Culling, W., Soil creep and the development of hillside slopes, *The Journal of Geology*, 71(2), 127–161, doi:<https://doi.org/10.1086/626891>, 1963.
- Culling, W., Theory of erosion on soil-covered slopes, *The Journal of Geology*, 73(2), 230–254, doi:<https://doi.org/10.1086/627060>, 1965.
- Czirók, A., E. Somfai, and T. Vicsek, Experimental Evidence for Self-Affine Roughening in a Micromodel of Geomorphological Evolution, *Physical Review Letters*, 71(13), 2154–2157, doi:[10.1103/PhysRevLett.71.2154](https://doi.org/10.1103/PhysRevLett.71.2154), 1993.
- Dai, Z., S. Fagherazzi, X. Mei, and J. Gao, Decline in Suspended Sediment Concentration Delivered by the Changjiang (Yangtze) River into the East China Sea between 1956 and 2013, *Geomorphology*, 268, 123–132, doi:[10.1016/j.geomorph.2016.06.009](https://doi.org/10.1016/j.geomorph.2016.06.009), 2016.
- Davis, W. M., The Convex Profile of Bad-Land Divides, *Science*, 20(508), 245–245, doi:[10.1126/science.ns-20.508.245](https://doi.org/10.1126/science.ns-20.508.245), 1892.
- De Girolamo, A. M., G. Pappagallo, and A. Lo Porto, Temporal Variability of Suspended Sediment Transport and Rating Curves in a Mediterranean River Basin: The Celone (SE Italy), *Catena*, 128, 135–143, doi:[10.1016/j.catena.2014.09.020](https://doi.org/10.1016/j.catena.2014.09.020), 2015.
- De Roo, A., C. Wesseling, and C. Ritsema, LISEM: A Single-Event Physically Based Hydrological and Soil Erosion Model for Drainage Basins. I: Theory, Input and Output, *Hydrological processes*, 10(8), 1107–1117, 1996.
- Dean, D. J., D. J. Topping, J. C. Schmidt, R. E. Griffiths, and T. A. Sabol, Sediment Supply versus Local Hydraulic Controls on Sediment Transport and Storage in a River with Large Sediment Loads, *Journal of Geophysical Research F: Earth Surface*, 121(1), 82–110, doi:[10.1002/2015JF003436](https://doi.org/10.1002/2015JF003436), 2016.
- Delon, G., D. Terwagne, S. Dorbolo, N. Vandewalle, and H. Caps, Impact of Liquid Droplets on Granular Media, *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 84(4), 3–7, doi:[10.1103/PhysRevE.84.046320](https://doi.org/10.1103/PhysRevE.84.046320), 2011.
- Densmore, A. L., M. A. Ellis, and R. S. Anderson, Landsliding and the Evolution of Normal-Fault-Bounded Mountains, *Journal of Geophysical Research: Solid Earth*, 103(B7), 15,203–15,219, doi:[10.1029/98JB00510](https://doi.org/10.1029/98JB00510), 1998.

Bibliography

- Ding, J., E. A. Johnson, and Y. E. Martin, Linking Soil Moisture Variation and Abundance of Plants to Geomorphic Processes: A Generalized Model for Erosion-Uplifting Landscapes, *Journal of Geophysical Research: Biogeosciences*, 123(3), 960–975, doi:10.1002/2017JG004244, 2018.
- Dodds, P. S., and D. H. Rothman, Scaling, Universality, and Geomorphology, *Annual Review of Earth and Planetary Sciences*, 28(1), 571–610, doi:10.1146/annurev.earth.28.1.571, 2000.
- Dodds, P. S., and D. H. Rothman, Geometry of River Networks. I. Scaling, Fluctuations, and Deviations, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 63(1), 016,115, doi:10.1103/PhysRevE.63.016115, 2001a.
- Dodds, P. S., and D. H. Rothman, Geometry of River Networks. II. Distributions of Component Size and Number, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 63(1), 016,116, doi:10.1103/PhysRevE.63.016116, 2001b.
- Dodds, P. S., and D. H. Rothman, Geometry of River Networks. III. Characterization of Component Connectivity, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 63(1), 016,117, doi:10.1103/PhysRevE.63.016117, 2001c.
- Dong, W., Q. Wang, B. Zhou, and Y. Shan, A Simple Model for the Transport of Soil-Dissolved Chemicals in Runoff by Raindrops, *Catena*, 101, 129–135, doi:10.1016/j.catena.2012.10.007, 2013.
- Dorrell, R. M., L. A. Amy, J. Peakall, and W. D. McCaffrey, Particle Size Distribution Controls the Threshold between Net Sediment Erosion and Deposition in Suspended Load Dominated Flows, *Geophysical Research Letters*, 45(3), 1443–1452, doi:10.1002/2017GL076489, 2018.
- Dunne, T., Formation and Controls of Channel Networks, *Progress in Physical Geography: Earth and Environment*, 4(2), 211–239, doi:10.1177/030913338000400204, 1980.
- Dunne, T., and B. F. Aubry, Evaluation of Horton's Theory of Sheetwash and Rill Erosion on the Basis of Field Experiments, in *Hillslope Processes*, pp. 31–53, George Allen & Unwin, London, UK, 1986.

- Dunne, T., D. V. Malm, and S. M. Mudd, A rain splash transport equation assimilating field and laboratory measurements, *Journal of Geophysical Research: Earth Surface*, 115(F1), F01,001, doi:10.1029/2009JF001302, 2010.
- Eder, A., P. Strauss, T. Krueger, and J. N. Quinton, Comparative Calculation of Suspended Sediment Loads with Respect to Hysteresis Effects (in the Petzenkirchen Catchment, Austria), *Journal of Hydrology*, 389(1-2), 168–176, doi:10.1016/j.jhydrol.2010.05.043, 2010.
- Edmaier, K., B. Crouzy, and P. Perona, Experimental Characterization of Vegetation Uprooting by Flow, *Journal of Geophysical Research: Biogeosciences*, 120(9), 1812–1824, doi:10.1002/2014JG002898, 2015.
- Edwards, S. F., and D. R. Wilkinson, The Surface Statistics of a Granular Aggregate, *Proc. R. Soc. Lond. A*, 381(1780), 17–31, doi:10.1098/rspa.1982.0056, 1982.
- Emady, H. N., D. Kayrak-Talay, W. C. Schwerin, and J. D. Litster, Granule Formation Mechanisms and Morphology from Single Drop Impact on Powder Beds, *Powder Technology*, 212(1), 69–79, doi:10.1016/j.powtec.2011.04.030, 2011.
- Erskine, R. H., T. R. Green, J. A. Ramirez, and L. H. MacDonald, Comparison of grid-based algorithms for computing upslope contributing area, *Water Resources Research*, 42(9), doi:10.1029/2005WR004648, 2006.
- Fatichi, S., et al., An Overview of Current Applications, Challenges, and Future Trends in Distributed Process-Based Models in Hydrology, *Journal of Hydrology*, 537, 45–60, doi:10.1016/j.jhydrol.2016.03.026, 2016.
- Feng, T., W. Wei, L. Chen, J. Rodrigo-Comino, C. Die, X. Feng, K. Ren, E. C. Brevik, and Y. Yu, Assessment of the Impact of Different Vegetation Patterns on Soil Erosion Processes on Semiarid Loess Slopes, *Earth Surface Processes and Landforms*, 0(0), doi:10.1002/esp.4361, 2018.
- Ferdowsi, B., C. P. Ortiz, and D. J. Jerolmack, Glassy Dynamics of Landscape Evolution, *Proceedings of the National Academy of Sciences*, p. 201715250, doi:10.1073/pnas.1715250115, 2018.
- Flanagan, D., and M. Nearing, USDA-Water Erosion Prediction Project: Hillslope Profile and Watershed Model Documentation, *Tech. rep.*, NSERL report, 1995.

Bibliography

- Flanagan, D. C., J. C. Ascough, M. A. Nearing, and J. M. Laflen, The Water Erosion Prediction Project (WEPP) Model, in *Landscape Erosion and Evolution Modeling*, pp. 145–199, Springer, 2001.
- Flint, J.-J., Experimental Development of Headward Growth of Channel Networks, *GSA Bulletin*, 84(3), 1087–1094, doi:10.1130/0016-7606(1973)84<1087:EDOHGO>2.0.CO;2, 1973.
- Forte, A. M., B. J. Yanites, and K. X. Whipple, Complexities of landscape evolution during incision through layered stratigraphy with contrasts in rock strength, *Earth Surface Processes and Landforms*, 41(12), 1736–1757, doi:10.1002/esp.3947, 2016.
- Foufoula-Georgiou, E., and C. Stark, Introduction to Special Section on Stochastic Transport and Emergent Scaling on Earth's Surface: Rethinking Geomorphic Transport—Stochastic Theories, Broad Scales of Motion and Nonlocality, *Journal of Geophysical Research: Earth Surface*, 115(F2), doi:10.1029/2010JF001661, 2010.
- Foufoula-Georgiou, E., V. Ganti, and W. E. Dietrich, A Nonlocal Theory of Sediment Transport on Hillslopes, *Journal of Geophysical Research: Earth Surface*, 115(F2), doi:10.1029/2009JF001280, 2010.
- Francipane, A., S. Fatichi, V. Y. Ivanov, and L. V. Noto, Stochastic assessment of climate impacts on hydrology and geomorphology of semiarid headwater basins using a physically based model, *Journal of Geophysical Research: Earth Surface*, 120(3), 507–533, doi:10.1002/2014JF003232, 2015.
- Franke, R., and G. Nielson, Smooth Interpolation of Large Sets of Scattered Data, *International Journal for Numerical Methods in Engineering*, 15(11), 1691–1704, doi:10.1002/nme.1620151110, 1980.
- Freeman, T., Calculating catchment area with divergent flow based on a regular grid, *Computers & Geosciences*, 17(3), 413–422, doi:10.1016/0098-3004(91)90048-I, 1991.
- Frigg, R., Self-Organised Criticality - What It Is and What It Isn't, *Studies in History and Philosophy of Science Part C :Studies in History and Philosophy of Biological and Biomedical Sciences*, 34(3), 613–632, doi:10.1016/S0039-3681(03)00046-3, 2003.
- Furbish, D. J., and J. J. Roering, Sediment Disentrainment and the Concept of Local versus Nonlocal Transport on Hillslopes, *Journal of Geophysical Research: Earth Surface*, 118(2), 937–952, doi:10.1002/jgrf.20071, 2013.

- Furbish, D. J., K. K. Hamner, M. Schmeeckle, M. N. Borosund, and S. M. Mudd, Rain Splash of Dry Sand Revealed by High-Speed Imaging and Sticky Paper Splash Targets, *Journal of Geophysical Research*, 112(F1), F01,001, doi:10.1029/2006JF000498, 2007.
- Gabet, E. J., and M. K. Mendoza, Particle Transport over Rough Hillslope Surfaces by Dry Ravel: Experiments and Simulations with Implications for Nonlocal Sediment Flux, *Journal of Geophysical Research: Earth Surface*, 117(F1), doi:10.1029/2011JF002229, 2012.
- Gao, B., M. Walter, T. Steenhuis, J.-Y. Parlange, K. Nakano, C. Rose, and W. Hogarth, Investigating Ponding Depth and Soil Detachability for a Mechanistic Erosion Model Using a Simple Experiment, *Journal of Hydrology*, 277(1-2), 116–124, doi:10.1016/S0022-1694(03)00085-4, 2003.
- Gao, B., M. T. Walter, T. S. Steenhuis, J. Y. Parlange, B. K. Richards, W. L. Hogarth, and C. W. Rose, Investigating Raindrop Effects on Transport of Sediment and Non-Sorbed Chemicals from Soil to Surface Runoff, *Journal of Hydrology*, 308(1), 313–320, doi:10.1016/j.jhydrol.2004.11.007, 2005.
- Gao, P., and M. Josefson, Event-Based Suspended Sediment Dynamics in a Central New York Watershed, *Geomorphology*, 139-140, 425–437, doi:10.1016/j.geomorph.2011.11.007, 2012.
- Ghahramani, A., and Y. Ishikawa, Water Flux and Sediment Transport within a Forested Landscape: The Role of Connectivity, Subsurface Flow, and Slope Length Scale on Transport Mechanism, *Hydrological Processes*, 27(26), 4091–4102, doi:10.1002/hyp.9791, 2013.
- Ghahramani, A., I. Yoshiharu, and S. M. Mudd, Field Experiments Constraining the Probability Distribution of Particle Travel Distances during Natural Rainstorms on Different Slope Gradients, *Earth Surface Processes and Landforms*, 37(5), 473–485, doi:10.1002/esp.2253, 2012.
- Giacometti, A., A. Maritan, and J. R. Banavar, Continuum Model for River Networks, *Physical Review Letters*, 75(3), 577–580, doi:10.1103/PhysRevLett.75.577, 1995.
- Gilbert, G. K., Land Sculptures - Geology of the Henry Mountains, *Monograph*, pp. i–160, doi:10.1038/022266c0, 1877.

Bibliography

- Glock, W. S., The Development of Drainage Systems: A Synoptic View, *Geographical Review*, 21(3), 475–482, doi:10.2307/209434, 1931.
- Gomez, B., and V. T. Mullen, An Experimental Study of Sapped Drainage Network Development, *Earth Surface Processes and Landforms*, 17(5), 465–476, doi:https://doi.org/10.1002/esp.3290170506, 1992.
- Gómez, J. A., F. Darboux, and M. A. Nearing, Development and Evolution of Rill Networks under Simulated Rainfall, *Water Resources Research*, 39(6), 1148, doi:10.1029/2002WR001437, 2003.
- Gordon, L. M., S. J. Bennett, and R. R. Wells, Evolution of Rill Networks on Soil - Mantled Experimental Landscapes Driven by Rainfall and Baselevel Adjustments, *Landform Analysis*, 17, 57–63, 2011.
- Gordon, L. M., S. J. Bennett, and R. R. Wells, Response of a Soil-Mantled Experimental Landscape to Exogenic Forcing, *Water Resources Research*, 48(10), W10,514, doi:10.1029/2012WR012283, 2012.
- Goren, L., S. D. Willett, F. Herman, and J. Braun, Coupled Numerical-Analytical Approach to Landscape Evolution Modeling, *Earth Surface Processes and Landforms*, 39(4), 522–545, doi:10.1002/esp.3514, 2014.
- Graveleau, F., J. Malavieille, and S. Dominguez, Experimental Modelling of Orogenic Wedges: A Review, *Tectonophysics*, 538, 1–66, doi:10.1016/j.tecto.2012.01.027, 2012.
- Gray, D. M., Interrelationships of Watershed Characteristics, *Journal of Geophysical Research*, 66(4), 1215–1223, doi:10.1029/JZ066i004p01215, 1961.
- Hack, J., Studies of Longitudinal Stream Profiles in Virginia and Maryland, *Report 294B*, uSGS Publications Warehouse, 1957.
- Hadka, D., and P. Reed, Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework, *Evolutionary Computation*, 21(2), 231–259, doi:10.1162/EVCO_a_00075, 2013.
- Hairsine, P., and C. Rose, Rainfall Detachment and Deposition: Sediment Transport in the Absence of Flow-Driven Processes, *Soil Science Society of America Journal*, 55(2), 320, doi:10.2136/sssaj1991.03615995005500020003x, 1991.

- Hairsine, P. B., and C. W. Rose, Modeling Water Erosion Due to Overland Flow Using Physical Principles: 2. Rill Flow, *Water Resources Research*, 28(1), 245–250, doi:10.1029/91WR02381, 1992a.
- Hairsine, P. B., and C. W. Rose, Modeling Water Erosion Due to Overland Flow Using Physical Principles: 1. Sheet Flow, *Water Resources Research*, 28(1), 237–243, doi:10.1029/91WR02380, 1992b.
- Hairsine, P. B., G. C. Sander, C. W. Rose, J.-Y. Y. Parlange, W. L. Hogarth, I. Lisle, and H. Rouhipour, Unsteady Soil Erosion Due to Rainfall Impact: A Model of Sediment Sorting on the Hillslope, *Journal of Hydrology*, 220(3-4), 115–128, doi:10.1016/S0022-1694(99)00068-2, 1999.
- Halliday, S. J., R. A. Skeffington, M. J. Bowes, E. Gozzard, J. R. Newman, M. Loewenthal, E. J. Palmer-Felgate, H. P. Jarvie, and A. J. Wade, The Water Quality of the River Enborne, UK: Observations from High-Frequency Monitoring in a Rural, Lowland River System, *Water*, 6(1), 150–180, doi:10.3390/w6010150, 2014.
- Hamlett, C. A. E., S. Atherton, N. J. Shirtcliffe, G. McHale, S. Ahn, S. H. Doerr, R. Bryant, and M. I. Newton, Transitions of Water-Drop Impact Behaviour on Hydrophobic and Hydrophilic Particles, *European Journal of Soil Science*, 64(3), 324–333, doi:10.1111/ejss.12003, 2013.
- Hamshaw, S. D., M. M. Dewoolkar, A. W. Schroth, B. C. Wemple, and D. M. Rizzo, A New Machine-Learning Approach for Classifying Hysteresis in Suspended-Sediment Discharge Relationships Using High-Frequency Monitoring Data, *Water Resources Research*, 54(6), 4040–4058, doi:10.1029/2017WR022238, 2018.
- Han, J., N. M. Gasparini, and J. P. L. Johnson, Measuring the imprint of orographic rainfall gradients on the morphology of steady-state numerical fluvial landscapes, *Earth Surface Processes and Landforms*, 40(10), 1334–1350, doi:10.1002/esp.3723, 2015.
- Hancock, G., and G. Willgoose, Use of a Landscape Simulator in the Validation of the SIBERIA Catchment Evolution Model: Declining Equilibrium Landforms, *Water Resources Research*, 37(7), 1981–1992, doi:10.1029/2001WR900002, 2001.
- Hancock, G., D. Verdon-Kidd, and J. Lowry, Soil erosion predictions from a landscape evolution model – An assessment of a post-mining landform using spatial climate

Bibliography

- change analogues, *Science of The Total Environment*, 601, 109–121, doi:10.1016/j.scitotenv.2017.04.038, 2017.
- Hancock, G. R., and G. R. Willgoose, The Use of a Landscape Simulator in the Validation of the Siberia Landscape Evolution Model: Transient Landforms, *Earth Surface Processes and Landforms*, 27(12), 1321–1334, doi:10.1002/esp.414, 2002.
- Hasbargen, L. E., and C. Paola, Landscape Instability in an Experimental Drainage Basin, *Geology*, 28(12), 1067–1070, 2000.
- Hasbargen, L. E., and C. Paola, How Predictable Is Local Erosion Rate in Eroding Landscapes?, *Prediction in Geomorphology*, pp. 231–240, doi:10.1029/135GM16, 2003.
- He, J., X. Li, L. Jia, H. Gong, and Q. Cai, Experimental Study of Rill Evolution Processes and Relationships between Runoff and Erosion on Clay Loam and Loess, *Soil Science Society of America Journal*, 78(5), 1716–1725, 2014.
- Heilig, A., et al., Testing a Mechanistic Soil Erosion Model with a Simple Experiment, *Journal of Hydrology*, 244(1-2), 9–16, doi:10.1016/S0022-1694(00)00400-5, 2001.
- Helmlinger, K. R., P. Kumar, and E. Foufoula-Georgiou, On the Use of Digital Elevation Model Data for Hortonian and Fractal Analyses of Channel Networks, *Water Resources Research*, 29(8), 2599–2613, doi:10.1029/93WR00545, 1993.
- Heng, B. C. P., G. C. Sander, and C. F. Scott, Modeling Overland Flow and Soil Erosion on Nonuniform Hillslopes: A Finite Volume Scheme, *Water Resources Research*, 45(5), doi:10.1029/2008WR007502, 2009.
- Heng, B. C. P., G. C. Sander, A. Armstrong, J. N. Quinton, J. H. Chandler, and C. F. Scott, Modeling the Dynamics of Soil Erosion and Size-Selective Sediment Transport over Nonuniform Topography in Flume-Scale Experiments, *Water Resources Research*, 47(2), doi:10.1029/2010WR009375, 2011.
- Herman, F., O. Beyssac, M. Brughelli, S. N. Lane, S. Leprince, T. Adatte, J. Y. Y. Lin, J.-P. Avouac, and S. C. Cox, Erosion by an Alpine Glacier, *Science*, 350(6257), 193–195, doi:10.1126/science.aab2386, 2015.

- Horton, R. E., Erosional Development of Streams and Their Drainage Basins; Hydrophysical Approach to Quantitative Morphology, *Geological Society of America Bulletin*, 56(3), 275, doi:10.1130/0016-7606(1945)56[275:EDOSAT]2.0.CO;2, 1945.
- Houssais, M., C. P. Ortiz, D. J. Durian, and D. J. Jerolmack, Onset of Sediment Transport Is a Continuous Transition Driven by Fluid Shear and Granular Creep, *Nature Communications*, 6, 6527, doi:10.1038/ncomms7527, 2015.
- Howard, A., W. Dietrich, and M. A. Seidl, Modeling Fluvial Erosion on Regional to Continental Scales, *Journal of Geophysical Research*, 99, 13,971–1398, doi:10.1029/94JB00744, 1994.
- Howard, A. A. D., Modelling Fluvial Systems: Rock-, Gravel-, and Sand-Bed Channels, in *River Channels: Environment and Process*, richards, k. s. ed., pp. 69–94, Blackwell Science, Blackwell, Oxford, UK, 1987.
- Howard, A. D., A detachment-limited model of drainage basin evolution, *Water Resources Research*, 30(7), 2261, doi:10.1029/94WR00757, 1994.
- Hu, F., J. Liu, C. Xu, W. Du, Z. Yang, X. Liu, G. Liu, and S. Zhao, Soil Internal Forces Contribute More than Raindrop Impact Force to Rainfall Splash Erosion, *Geoderma*, 330, 91–98, doi:10.1016/j.geoderma.2018.05.031, 2018.
- Hunt, A. G., Brief Communication: Possible Explanation of the Values of Hack's Drainage Basin, River Length Scaling Exponent, *Nonlinear Processes in Geophysics*, 23(2), 91–93, doi:https://doi.org/10.5194/npg-23-91-2016, 2016.
- Istanbulluoglu, E., and R. L. Bras, Vegetation-modulated landscape evolution: Effects of vegetation on landscape processes, drainage density, and topography, *Journal of Geophysical Research: Earth Surface*, 110(2), F02,012, doi:10.1029/2004JF000249, 2005.
- Jaafari, A., A. Najafi, J. Rezaeian, and A. Sattarian, Modeling Erosion and Sediment Delivery from Unpaved Roads in the North Mountainous Forest of Iran, *GEM - International Journal on Geomathematics*, 6(2), 343–356, doi:10.1007/s13137-014-0062-4, 2015.
- Jaberi, M., M. R. Ghassemi, S. Shayan, M. Yamani, and S. M. Zamanzadeh, Interaction between Active Tectonics, Erosion and Diapirism, a Case Study from Habbale-Rud in

Bibliography

- Southern Central Alborz (Northern Iran), *Geomorphology*, 300, 77–94, doi:10.1016/j.geomorph.2017.01.031, 2018.
- Jafari, R., and L. Bakhshandehmehr, Quantitative Mapping and Assessment of Environmentally Sensitive Areas to Desertification in Central Iran, *Land Degradation & Development*, 27(2), 108–119, doi:10.1002/ldr.2227, 2013.
- Jajcevic, D., E. Siegmund, C. Radeke, and J. G. Khinast, Large-Scale CFD–DEM Simulations of Fluidized Granular Systems, *Chemical Engineering Science*, 98, 298–310, doi:10.1016/j.ces.2013.05.014, 2013.
- Jeffery, M. L., B. J. Yanites, C. J. Poulsen, and T. A. Ehlers, Vegetation-precipitation controls on Central Andean topography, *Journal of Geophysical Research: Earth Surface*, 119(6), 1354–1375, doi:10.1002/2013JF002919, 2014.
- Jomaa, S., D. Barry, A. Brovelli, G. Sander, J.-Y. Parlange, B. Heng, and H. Tromp-van Meerveld, Effect of Raindrop Splash and Transversal Width on Soil Erosion: Laboratory Flume Experiments and Analysis with the Hairsine–Rose Model, *Journal of Hydrology*, 395(1-2), 117–132, doi:10.1016/J.JHYDROL.2010.10.021, 2010.
- Jomaa, S., D. A. Barry, A. Brovelli, B. C. P. Heng, G. C. Sander, J. Y. Parlange, and C. W. Rose, Rain Splash Soil Erosion Estimation in the Presence of Rock Fragments, *Catena*, 92, 38–48, doi:10.1016/j.catena.2011.11.008, 2012a.
- Jomaa, S., D. A. Barry, B. C. P. Heng, A. Brovelli, G. C. Sander, and J. Y. Parlange, Influence of Rock Fragment Coverage on Soil Erosion and Hydrological Response: Laboratory Flume Experiments and Modeling, *Water Resources Research*, 48(5), doi:10.1029/2011WR011255, 2012b.
- Jomaa, S., D. A. Barry, B. C. P. Heng, A. Brovelli, G. C. Sander, and J. Y. Parlange, Effect of Antecedent Conditions and Fixed Rock Fragment Coverage on Soil Erosion Dynamics through Multiple Rainfall Events, *Journal of Hydrology*, 484, 115–127, doi:10.1016/j.jhydrol.2013.01.021, 2013.
- Joung, Y. S., and C. R. Buie, Aerosol Generation by Raindrop Impact on Soil, *Nature Communications*, 6, 6083, doi:10.1038/ncomms7083, 2015.
- Juez, C., M. A. Hassan, and M. J. Franca, The Origin of Fine Sediment Determines the Observations of Suspended Sediment Fluxes Under Unsteady Flow Conditions, *Water Resources Research*, 54(8), 5654–5669, doi:10.1029/2018WR022982, 2018.

- Karaca, S., F. Gülser, and R. Selçuk, Relationships between soil properties, topography and land use in the Van Lake Basin, Turkey, *Eurasian Journal of Soil Science*, 7(2), 115–120, doi:10.18393/ejss.348412, 2018.
- Karimae Tabarestani, M., and A. R. Zarrati, Sediment Transport during Flood Event: A Review, *International Journal of Environmental Science and Technology*, 12(2), 775–788, doi:10.1007/s13762-014-0689-6, 2015.
- Katsuragi, H., Morphology Scaling of Drop Impact onto a Granular Layer, *Physical Review Letters*, 104(21), 1–4, doi:10.1103/PhysRevLett.104.218001, 2010.
- Katsuragi, H., Length and Time Scales of a Liquid Drop Impact and Penetration into a Granular Layer, *Journal of Fluid Mechanics*, 675(2011), 552–573, doi:10.1017/jfm.2011.31, 2011.
- Katsuragi, H., *Physics of Soft Impact and Cratering*, Springer, Tokyo, Japan, doi:10.1007/978-4-431-55648-0, 2016.
- Keesstra, S. D., J. Maroulis, E. Argaman, A. Voogt, and L. Wittenberg, Effects of Controlled Fire on Hydrology and Erosion under Simulated Rainfall, *Cuadernos de Investigación Geográfica*, 40(2), 269–293, doi:10.18172/cig.2532, 2014.
- Kennedy, J., Particle Swarm Optimization, *Encyclopedia of Machine Learning*, 46(11), 685–691, doi:10.1109/ICNN.1995.488968, 2010.
- Khalili Moghadam, B., M. Jabarifar, M. Bagheri, and E. Shahbazi, Effects of Land Use Change on Soil Splash Erosion in the Semi-Arid Region of Iran, *Geoderma*, 241–242, 210–220, doi:10.1016/j.geoderma.2014.11.025, 2015.
- Kiani-Harchegani, M., S. H. Sadeghi, and H. Asadi, Comparing Grain Size Distribution of Sediment and Original Soil under Raindrop Detachment and Raindrop-Induced and Flow Transport Mechanism, *Hydrological Sciences Journal*, 63(2), 312–323, doi:10.1080/02626667.2017.1414218, 2018.
- Kim, J., and V. Y. Ivanov, On the Nonuniqueness of Sediment Yield at the Catchment Scale: The Effects of Soil Antecedent Conditions and Surface Shield, *Water Resources Research*, 50(2), 1025–1045, doi:10.1002/2013WR014580, 2014.

Bibliography

- Kim, J., V. Y. Ivanov, and N. D. Katopodes, Modeling Erosion and Sedimentation Coupled with Hydrological and Overland Flow Processes at the Watershed Scale, *Water Resources Research*, 49(9), 5134–5154, doi:10.1002/wrcr.20373, 2013.
- Kim, J., M. C. Dwelle, S. K. Kampf, S. Fatichi, and V. Y. Ivanov, On the Non-Uniqueness of the Hydro-Geomorphologic Responses in a Zero-Order Catchment with Respect to Soil Moisture, *Advances in Water Resources*, 92, 73–89, doi:10.1016/j.advwatres.2016.03.019, 2016a.
- Kim, J., V. Y. Ivanov, and S. Fatichi, Environmental Stochasticity Controls Soil Erosion Variability, *Scientific Reports*, 6, 22,065, doi:10.1038/srep22065, 2016b.
- Kinnell, P. I. A., Modeling of the Effect of Flow Depth on Sediment Discharged by Rain-Impacted Flows from Sheet and Interrill Erosion Areas: A Review, *Hydrological Processes*, 27(18), 2567–2578, doi:10.1002/hyp.9363, 2013.
- Kirkby, M., Measurement and Theory of Soil Creep, *The Journal of Geology*, 75(4), 359–378, doi:10.1086/627267, 1967.
- Kirkby, M. J., Hillslope Process-Response Models Based on the Continuity Equation, *Tech. Rep. 3*, Institute of British Geographers special Publications, London, UK, 1971.
- Kirkby, M. J., A Basis for Soil Profile Modelling in a Geomorphic Context, *Journal of Soil Science*, 36(1), 97–121, doi:10.1111/j.1365-2389.1985.tb00316.x, 1985.
- Klein, M., Anti Clockwise Hysteresis in Suspended Sediment Concentration during Individual Storms: Holbeck Catchment; Yorkshire, England, *Catena*, 11(2-3), 251–257, doi:10.1016/0341-8162(84)90014-6, 1984.
- Kloss, C., C. Goniva, A. Hager, S. Amberger, and S. Pirker, Models, Algorithms and Validation for Opensource DEM and CFD–DEM, *Progress in Computational Fluid Dynamics, an International Journal*, 12(2-3), 140–152, doi:10.1504/PCFD.2012.047457, 2012.
- Knighton, D., *Fluvial Forms and Processes: A New Perspective*, 2 ed., Routledge, London, UK, 2014.

- Lague, D., A. Crave, and P. Davy, Laboratory Experiments Simulating the Geomorphic Response to Tectonic Uplift, *Journal of Geophysical Research: Solid Earth*, 108(B1), 2008, doi:10.1029/2002JB001785, 2003.
- Lardier, N., P. Roudier, B. Clothier, and G. R. Willmott, High-Speed Photography of Water Drop Impacts on Sand and Soil, *European Journal of Soil Science*, doi:10.1111/ejss.12737, 2018.
- Lee, J. B., D. Derome, and J. Carmeliet, Drop Impact on Natural Porous Stones, *Journal of Colloid and Interface Science*, 469, 147–156, doi:10.1016/j.jcis.2016.02.008, 2016a.
- Lee, J. B., D. Derome, A. Dolatabadi, and J. Carmeliet, Energy Budget of Liquid Drop Impact at Maximum Spreading: Numerical Simulations and Experiments, *Langmuir*, 32(5), 1279–1288, doi:10.1021/acs.langmuir.5b03848, 2016b.
- Leenman, A., and J. Tunncliffe, Genesis of a Major Gully Mass-Wasting Complex, and Implications for Valley Filling, East Cape, New Zealand, *GSA Bulletin*, 130(7-8), 1121–1130, doi:10.1130/B31849.1, 2018.
- Legout, C., S. Legu  dois, Y. Le Bissonnais, and O. Malam Issa, Splash Distance and Size Distributions for Various Soils, *Geoderma*, 124(3-4), 279–292, doi:10.1016/j.geoderma.2004.05.006, 2005.
- Leopold, L. B., T. Maddock Jnr, and T. J. Maddock, The Hydraulic Geomtry of Stream Channels and Some Physiographic Implications, *Geological Survey Professional Paper* 252, p. 57, 1953.
- Lisle, I. G., C. W. Rose, W. L. Hogarth, P. B. Hairsine, G. C. Sander, and J. Y. Parlange, Stochastic Sediment Transport in Soil Erosion, *Journal of Hydrology*, 204(1-4), 217–230, doi:10.1016/S0022-1694(97)00123-6, 1998.
- Lisle, I. G., et al., Transport Time Scales in Soil Erosion Modeling, *Vadose Zone Journal*, 16(12), doi:10.2136/vzj2017.06.0121, 2017.
- Lloyd, C. E. M., J. E. Freer, P. J. Johnes, and A. L. Collins, Using Hysteresis Analysis of High-Resolution Water Quality Monitoring Data, Including Uncertainty, to Infer Controls on Nutrient and Sediment Transfer in Catchments, *Science of the Total Environment*, 543(2016), 388–404, doi:10.1016/j.scitotenv.2015.11.028, 2016.

Bibliography

- Long, E. J., G. K. Hargrave, J. R. Cooper, B. G. B. Kitchener, A. J. Parsons, C. J. M. Hewett, and J. Wainwright, Experimental Investigation into the Impact of a Liquid Droplet onto a Granular Bed Using Three-Dimensional, Time-Resolved, Particle Tracking, *Physical Review E*, 89(3), 032,201, doi:10.1103/PhysRevE.89.032201, 2014.
- Lord Rayleigh, On the Flow of Viscous Liquids, Especially in Two Dimensions, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 36(221), 354–372, doi:10.1080/14786449308620489, 1893.
- Lovejoy, S., D. Schertzer, and A. A. Tsonis, Functional Box-Counting and Multiple Elliptical Dimensions in Rain, *Science*, 235(4792), 1036–1038, doi:10.1126/science.235.4792.1036, 1987.
- Luding, S., Introduction to Discrete Element Methods, *European Journal of Environmental and Civil Engineering*, 12(7-8), 785–826, doi:10.1080/19648189.2008.9693050, 2008.
- Mahmoodabadi, M., and S. A. Sajjadi, Effects of rain intensity, slope gradient and particle size distribution on the relative contributions of splash and wash loads to rain-induced erosion, *Geomorphology*, 253, 159–167, doi:10.1016/j.geomorph.2015.10.010, 2016.
- Malusà, M. G., M. L. Frezzotti, S. Ferrando, E. Brandmayr, F. Romanelli, and G. F. Panza, Active Carbon Sequestration in the Alpine Mantle Wedge and Implications for Long-Term Climate Trends, *Scientific Reports*, 8(1), 4740, doi:10.1038/s41598-018-22877-7, 2018.
- Mandelbrot, B. B., *The Fractal Geometry of Nature*, W. H. Freeman and Company, New York, USA, 1982.
- Marani, A., R. Rigon, and A. Rinaldo, A Note on Fractal Channel Networks, *Water Resources Research*, 27(12), 3041–3049, doi:10.1029/91WR02077, 1991.
- Marchamalo, M., J. M. Hooke, and P. J. Sandercock, Flow and Sediment Connectivity in Semi-Arid Landscapes in SE Spain: Patterns and Controls, *Land Degradation and Development*, 27(4), 1032–1044, doi:10.1002/ldr.2352, 2015.
- Marengo, M., C. Antonini, I. V. Roisman, and C. Tropea, Drop Collisions with Simple and Complex Surfaces, *Current Opinion in Colloid and Interface Science*, 16(4), 292–302, doi:10.1016/j.cocis.2011.06.009, 2011.

- Maritan, A., A. Rinaldo, R. Rigon, A. Giacometti, I. Rodriguez-Iturbe, and I. Rodríguez-Iturbe, Scaling Laws for River Networks, *Physical Review E*, 53(2), 1510–1515, doi:10.1103/PhysRevE.53.1510, 1996.
- Marković, D., and C. Gros, Power Laws and Self-Organized Criticality in Theory and Nature, *Physics Reports*, 536(2), 41–74, doi:10.1016/j.physrep.2013.11.002, 2014.
- Marston, J. O., S. T. Thoroddsen, W. K. Ng, and R. B. H. Tan, Experimental Study of Liquid Drop Impact onto a Powder Surface, *Powder Technology*, 203(2), 223–236, doi:10.1016/j.powtec.2010.05.012, 2010.
- Martin, Y., Modelling hillslope evolution: Linear and nonlinear transport relations, *Geomorphology*, 34(1-2), 1–21, doi:10.1016/S0169-555X(99)00127-0, 2000.
- Masselink, R. J. H., S. D. Keesstra, A. J. A. M. Temme, M. Seeger, R. Giménez, and J. Casalí, Modelling Discharge and Sediment Yield at Catchment Scale Using Connectivity Components, *Land Degradation & Development*, 27(4), 933–945, doi:10.1002/ldr.2512, 2016.
- Massong, T. M., and D. R. Montgomery, Influence of sediment supply, lithology, and wood debris on the distribution of bedrock and alluvial channels, *Geological Society of America Bulletin*, 112(4), 591–599, doi:10.1130/0016-7606(2000)112<591:IOSSLA>2.0.CO;2, 2000.
- Matsushita, M., and S. Ouchi, On the Self-Affinity of Various Curves, *Physica D: Nonlinear Phenomena*, 38(1-3), 246–251, 1989.
- McGuire, L. A., J. D. Pelletier, J. A. Gómez, and M. A. Nearing, Controls on the Spacing and Geometry of Rill Networks on Hillslopes: Rain Splash Detachment, Initial Hillslope Roughness, and the Competition between Fluvial and Colluvial Transport, *Journal of Geophysical Research: Earth Surface*, 118(1), 241–256, doi:10.1002/jgrf.20028, 2013.
- Meinhardt, H., Models and Hypotheses, *Differentiation*, 6(1-3), 117–123, doi:10.1111/j.1432-0436.1976.tb01478.x, 1976.
- Meinhardt, H., Models of biological pattern formation, (March), [http://citeseerx.ist.psu.edu/viewdoc/download?doi=](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1016/S0070-2153(07)81001-5), doi:10.1016/S0070-2153(07)81001-5, 1982.

Bibliography

- Miller, W. P., and M. K. Baharuddin, Particle Size of Interrill-Eroded Sediments from Highly Weathered Soils, *Soil Science Society of America Journal*, 51(6), 1610, doi:10.2136/sssaj1987.03615995005100060037x, 1987.
- Mock, S. J., A Classification Channel Links in Stream Networks, *Water Resources Research*, 7(6), 1558–1566, doi:10.1029/WR007i006p01558, 1971.
- Mohammad, A. G., and M. A. Adam, The Impact of Vegetative Cover Type on Runoff and Soil Erosion under Different Land Uses, *Catena*, 81(2), 97–103, doi:10.1016/j.catena.2010.01.008, 2010.
- Montgomery, D. R., and W. E. Dietrich, Channel Initiation and the Problem of Landscape Scale., *Science*, 255(5046), 826–830, doi:10.1126/science.255.5046.826, 1992.
- Morgan, R., J. Quinton, R. Smith, G. Govers, J. Poesen, K. Auerswald, G. Chisci, D. Torri, and M. Styczen, The European Soil Erosion Model (EUROSEM): A Dynamic Approach for Predicting Sediment Transport from Fields and Small Catchments, *Earth surface processes and landforms*, 23(6), 527–544, 1998.
- Morgan, R. P. C., and M. Nearing, *Handbook of Erosion Modelling*, John Wiley & Sons, Oxford, UK, 10.1002/9781444328455, 2011.
- Morvan, X., L. Verbeke, S. Laratte, and A. R. Schneider, Impact of Recent Conversion to Organic Farming on Physical Properties and Their Consequences on Runoff, Erosion and Crusting in a Silty Soil, *Catena*, 165, 398–407, doi:10.1016/j.catena.2018.02.024, 2018.
- Mosley, M. P., Experimental Study of Rill Erosion, *Transactions of the ASAE*, 17(5), 0909–0913, doi:10.13031/2013.36996, 1974.
- Mosley, M. P., and R. S. Parker, Re-Evaluation of the Relationship of Master Streams and Drainage Basins: Discussion, *Geological Society of America Bulletin*, 84(9), 3123, doi:10.1130/0016-7606(1973)84<3123:ROTRM>2.0.CO;2, 1973.
- Mudd, S. M., Detection of transience in eroding landscapes, *Earth Surface Processes and Landforms*, 42(1), 24–41, doi:10.1002/esp.3923, 2016.

- Mueller, J. E., Re-Evaluation of the Relationship of Master Streams and Drainage Basins, *Bulletin of the Geological Society of America*, 83(11), 3471–3474, doi:10.1130/0016-7606(1972)83[3471:ROTRROM]2.0.CO;2, 1972.
- Mueller, J. E., Re-Evaluation of the Relationship of Master Streams and Drainage Basins: Reply, *GSA Bulletin*, 84(9), doi:https://doi.org/10.1130/0016-7606(1973)84<3127:ROTRROM>2.0.CO;2, 1973.
- Nadal-Romero, E., D. Regüés, and J. Latron, Relationships among Rainfall, Runoff, and Suspended Sediment in a Small Catchment with Badlands, *Catena*, 74(2), 127–136, doi:10.1016/j.catena.2008.03.014, 2008.
- Naghdi, R., P. Dalir, V. Gholami, and H. R. Pourghasemi, Modeling of Sediment Generation from Forest Roads Employing SEDMODL and Its Calibration for Hyrcanian Forests in Northern Iran, *Environmental Earth Sciences*, 76(12), 414, doi:10.1007/s12665-017-6758-7, 2017.
- Neal, J. H., The Effect of the Degree of Slope and Rainfall Characteristics on Runoff and Soil Erosion, *Soil Science Society of America Journal*, 2(C), 525–532, doi:10.2136/sssaj1938.036159950002000C0083x, 1938.
- Nearing, M. A., G. R. Foster, L. J. Lane, and S. C. Finkner, A Process-Based Soil Erosion Model for USDA-Water Erosion Prediction Project Technology, *Transactions of the ASAE*, 32(5), 1587, doi:10.13031/2013.31195, 1989.
- Nearing, M. A., F. F. Pruski, and M. R. O’Neal, Expected Climate Change Impacts on Soil Erosion Rates: A Review, *Journal of Soil and Water Conservation*, 59(1), 43–50, 2004.
- Novara, A., A. Pisciotto, M. Minacapilli, A. Maltese, F. Capodici, A. Cerdà, and L. Gristina, The Impact of Soil Erosion on Soil Fertility and Vine Vigor. A Multidisciplinary Approach Based on Field, Laboratory and Remote Sensing Approaches, *Science of The Total Environment*, 622-623, 474–480, doi:10.1016/j.scitotenv.2017.11.272, 2018.
- O’Callaghan, J. F., and D. M. Mark, The Extraction of Drainage Networks from Digital Elevation Data, *Computer Vision, Graphics, and Image Processing*, 28(3), 323–344, doi:10.1016/S0734-189X(84)80011-0, 1984.

Bibliography

- O'Connell, S., and A. Siafarikas, Addison Disease: Diagnosis and Initial Management, *Australian Family Physician*, 39(11), 834–837, doi:10.1024/0301-1526.32.1.54, 2010.
- Oeurng, C., S. Sauvage, and J.-M. Sánchez-Pérez, Assessment of Hydrology, Sediment and Particulate Organic Carbon Yield in a Large Agricultural Catchment Using the SWAT Model, *Journal of Hydrology*, 401(3-4), 145–153, doi:10.1016/j.jhydrol.2011.02.017, 2011.
- Oliveto, G., D. Palma, and A. Di Domenico, Scaling Properties of Laboratory-Generated River Networks, Braunschweig, Germany, 2010.
- Paik, K., and P. Kumar, Power-Law Behavior in Geometric Characteristics of Full Binary Trees, *Journal of Statistical Physics*, 142(4), 862–878, doi:10.1007/s10955-011-0125-y, 2011.
- Panagos, P., et al., Global Rainfall Erosivity Assessment Based on High-Temporal Resolution Rainfall Records, *Scientific Reports*, 7(1), 4175, doi:10.1038/s41598-017-04282-8, 2017.
- Paola, C., K. Straub, D. Mohrig, and L. Reinhardt, The “Unreasonable Effectiveness” of Stratigraphic and Geomorphic Experiments, *Earth-Science Reviews*, 97(1), 1–43, doi:10.1016/j.earscirev.2009.05.003, 2009.
- Park, E., and E. M. Latrubesse, Surface water types and sediment distribution patterns at the confluence of mega rivers: The Solimões-Amazon and Negro Rivers junction, *Water Resources Research*, 51(8), 6197–6213, doi:10.1002/2014WR016757, 2015.
- Parker, R. A., Experimental study of basin evolution and its hydrologic implications, PhD thesis, Fort Collins, Colorado State University, <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA051839>, 1977.
- Parlange, J. Y., W. L. Hogarth, C. W. Rose, G. G. Sander, P. Hairsine, and I. Lisle, Addendum to Unsteady Soil Erosion Model, *Journal of Hydrology*, 217(1-2), 149–156, doi:10.1016/S0022-1694(99)00012-8, 1999.
- Passalacqua, P., F. Porté-Agel, E. Foufoula-Georgiou, and C. Paola, Application of Dynamic Subgrid-Scale Concepts from Large-Eddy Simulation to Modeling Landscape Evolution, *Water Resources Research*, 42(6), doi:10.1029/2006WR004879, 2006.

- Pastor-Satorras, R., and D. H. Rothman, Stochastic Equation for the Erosion of Inclined Topography, *Physical Review Letters*, 80(19), 4349–4352, doi:10.1103/PhysRevLett.80.4349, 1998.
- Patton, P. C., and S. A. Schumm, Gully Erosion, Northwestern Colorado: A Threshold Phenomenon, *Geology*, 3(2), 88–90, doi:10.1130/0091-7613(1975)3<88:GENCAT>2.0.CO;2, 1975.
- Pavlidis, G., and V. A. Tsihrintzis, Environmental Benefits and Control of Pollution to Surface Water and Groundwater by Agroforestry Systems: A Review, *Water Resources Management*, 32(1), 1–29, doi:10.1007/s11269-017-1805-4, 2018.
- Pazzaglia, F. J., et al., Geomorphology, active tectonics and landscape evolution in the Mid-Atlantic region, in *Field Guides*, vol. 40, pp. 109–169, Geological Society of America, doi:10.1130/2015.0040(06)\%0020, 2015.
- Pedrazzini, A., F. Humair, M. Jaboyedoff, and M. Tonini, Characterisation and spatial distribution of gravitational slope deformation in the Upper Rhone catchment (Western Swiss Alps), *Landslides*, 13(2), 259–277, doi:10.1007/s10346-015-0562-9, 2016.
- Pelletier, J. D., Drainage Basin Evolution in the Rainfall Erosion Facility: Dependence on Initial Conditions, *Geomorphology*, 53(1), 183–196, doi:10.1016/S0169-555X(02)00353-7, 2003.
- Pelletier, J. D., Fractal Behavior in Space and Time in a Simplified Model of Fluvial Landform Evolution, *Geomorphology*, 91(3), 291–301, doi:10.1016/j.geomorph.2007.04.015, 2007.
- Perona, P., et al., Biomass Selection by Floods and Related Timescales: Part 1. Experimental Observations, *Advances in Water Resources*, 39, 85–96, doi:10.1016/j.advwatres.2011.09.016, 2012.
- Perron, J. T., Climate and the Pace of Erosional Landscape Evolution, *Annual Review of Earth and Planetary Sciences*, 45(1), 561–591, doi:10.1146/annurev-earth-060614-105405, 2017.
- Perron, J. T., W. E. Dietrich, and J. W. Kirchner, Controls on the Spacing of First-Order Valleys, *Journal of Geophysical Research: Earth Surface*, 113(4), 1–21, doi:10.1029/2007JF000977, 2008.

Bibliography

- Perron, J. T., J. W. Kirchner, and W. E. Dietrich, Formation of Evenly Spaced Ridges and Valleys, *Nature*, 460(7254), 502–505, doi:10.1038/nature08174, 2009.
- Phillips, L. F., and S. A. Schumm, Effect of Regional Slope on Drainage Networks, *Geology*, 15(9), 813–816, doi:10.1130/0091-7613(1987)15<813:EORSOD>2.0.CO;2, 1987.
- Pietroni, J., J. Jarsjö, A. O. Romanchenko, and S. R. Chalov, Model Analyses of the Contribution of In-Channel Processes to Sediment Concentration Hysteresis Loops, *Journal of Hydrology*, 527, 576–589, doi:10.1016/j.jhydrol.2015.05.009, 2015.
- Pietsch, S., P. Kieckhefen, S. Heinrich, M. Müller, M. Schönherr, and F. Kleine Jäger, CFD-DEM Modelling of Circulation Frequencies and Residence Times in a Prismatic Spouted Bed, *Chemical Engineering Research and Design*, 132, 1105–1116, doi:10.1016/j.cherd.2018.01.013, 2018.
- Pimentel, D., et al., Environmental and Economic Costs of Soil Erosion and Conservation Benefits, *Science*, 267(5201), 1117–1123, doi:10.1126/science.267.5201.1117, 1995.
- Planchon, O., and F. Darboux, A Fast, Simple and Versatile Algorithm to Fill the Depressions of Digital Elevation Models, *Catena*, 46(2), 159–176, doi:10.1016/S0341-8162(01)00164-3, 2002.
- Poesen, J., K. Vandaele, and B. van Wesemael, Gully Erosion: Importance and Model Implications, in *Modelling Soil Erosion by Water*, NATO ASI Series, pp. 285–311, Springer, Berlin, Heidelberg, doi:10.1007/978-3-642-58913-3_22, 1998.
- Polyakov, V. O., and M. A. Nearing, Sediment Transport in Rill Flow under Deposition and Detachment Conditions, *Catena*, 51(1), 33–43, doi:10.1016/S0341-8162(02)00090-5, 2003.
- Pouyat, R. V., I. D. Yesilonis, J. Russell-Anelli, and N. K. Neerchal, Soil Chemical and Physical Properties That Differentiate Urban Land-Use and Cover Types, *Soil Science Society of America Journal*, 71(3), 1010–1019, doi:10.2136/sssaj2006.0164, 2007.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2007.

- Prosdocimi, M., A. Cerdà, and P. Tarolli, Soil Water Erosion on Mediterranean Vineyards: A Review, *Catena*, 141, 1–21, doi:10.1016/j.catena.2016.02.010, 2016.
- Prosser, I. P., J. Chappell, and R. Gillespie, Holocene Valley Aggradation and Gully Erosion in Headwater Catchments, South-Eastern Highlands of Australia, *Earth Surface Processes and Landforms*, 19(5), 465–480, doi:10.1002/esp.3290190507, 1994.
- Quinn, P., K. Beven, P. Chevallier, and O. Planchon, The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models, *Hydrological Processes*, 5(1), 59–79, doi:10.1002/hyp.3360050106, 1991.
- Quinton, J. N., G. Govers, K. V. Oost, and R. D. Bardgett, The Impact of Agricultural Soil Erosion on Biogeochemical Cycling, *Nature Geoscience*, 3(5), 311–314, doi:10.1038/ngeo838, 2010.
- Raff, D. A., J. A. Ramírez, and J. L. Smith, Hillslope Drainage Development with Time: A Physical Experiment, *Geomorphology*, 62(3), 169–180, doi:10.1016/j.geomorph.2004.02.011, 2004.
- Ranjbar, S., M. Hooshyar, A. Singh, and D. Wang, Quantifying Climatic Controls on River Network Branching Structure Across Scales, *Water Resources Research*, doi:10.1029/2018WR022853, 2018.
- Reinhardt, L., and M. A. Ellis, The Emergence of Topographic Steady State in a Perpetually Dynamic Self-Organized Critical Landscape, *Water Resources Research*, 51(7), 4986–5003, doi:10.1002/2014WR016223, 2015.
- Reusser, L., P. Bierman, and D. Rood, Quantifying human impacts on rates of erosion and sediment transport at a landscape scale, *Geology*, 43(2), 171–174, doi:10.1130/G36272.1, 2015.
- Richardson, L. F., The Problem of Contiguity: An Appendix to Statistics of Deadly Quarrels, in *Yearbook of the Society for General Systems Research*, vol. 6, pp. 140–187, Ann Arbor, Michigan, 1961.
- Rieke-Zapp, D. H., and M. A. Nearing, Slope Shape Effects on Erosion, *Soil Science Society of America Journal*, 69(5), 1463–1471, doi:10.2136/sssaj2005.0015, 2005.

Bibliography

- Rigon, R., I. Rodríguez-Iturbe, A. Maritan, A. Giacometti, D. G. Tarboton, and A. Rinaldo, On Hack's Law, *Water Resources Research*, 32(11), 3367–3374, doi:10.1029/96WR02397, 1996.
- Rigon, R., I. Rodríguez-Iturbe, and A. Rinaldo, Feasible Optimality Implies Hack's Law, *Water Resources Research*, 34(11), 3181–3189, doi:10.1029/98WR02287, 1998.
- Rinaldo, A., I. Rodríguez-Iturbe, R. Rigon, R. L. Bras, E. Ijjasz-Vasquez, and A. Marani, Minimum Energy and Fractal Structures of Drainage Networks, *Water Resources Research*, 28(9), 2183–2195, doi:10.1029/92WR00801, 1992.
- Rinaldo, A., I. Rodríguez-Iturbe, R. Rigon, E. Ijjasz-Vasquez, and R. L. Bras, Self-Organized Fractal River Networks, *Physical Review Letters*, 70(6), 822–825, doi:10.1103/PhysRevLett.70.822, 1993.
- Rinaldo, A., R. Rigon, J. R. Banavar, A. Maritan, and I. Rodríguez-Iturbe, Evolution and Selection of River Networks: Statics, Dynamics, and Complexity., *Proceedings of the National Academy of Sciences of the United States of America*, 111(7), 2417–24, doi:10.1073/pnas.1322700111, 2014.
- Rodrigo-Comino, J., Five Decades of Soil Erosion Research in “Terroir”. The State-of-the-Art, *Earth-Science Reviews*, 179, 436–447, doi:10.1016/j.earscirev.2018.02.014, 2018.
- Rodrigo Comino, J., et al., Quantitative Comparison of Initial Soil Erosion Processes and Runoff Generation in Spanish and German Vineyards, *Science of The Total Environment*, 565, 1165–1174, doi:10.1016/j.scitotenv.2016.05.163, 2016.
- Rodríguez-Iturbe, I., and A. Rinaldo, *Fractal River Basins: Chance and Self-Organization*, Cambridge University Press, UK, 1997.
- Rodríguez-Iturbe, I., A. Rinaldo, R. Rigon, R. L. Bras, E. Ijjasz-Vasquez, and A. Marani, Fractal Structures as Least Energy Patterns: The Case of River Networks, *Geophysical Research Letters*, 19(9), 889–892, doi:10.1029/92GL00938, 1992.
- Rodríguez-Lloveras, X., G. Bussi, F. Francés, E. Rodríguez-Caballero, A. Solé-Benet, M. Calle, and G. Benito, Patterns of runoff and sediment production in response to land-use changes in an ungauged Mediterranean catchment, *Journal of Hydrology*, 531, 1054–1066, doi:10.1016/j.jhydrol.2015.11.014, 2015.

- Rohais, S., S. Bonnet, and R. Eschard, Sedimentary Record of Tectonic and Climatic Erosional Perturbations in an Experimental Coupled Catchment-Fan System, *Basin Research*, 24(2), 198–212, doi:10.1111/j.1365-2117.2011.00520.x, 2012.
- Römkens, M. J. M., K. Helming, and S. N. Prasad, Soil Erosion under Different Rainfall Intensities, Surface Roughness, and Soil Water Regimes, *Catena*, 46(2), 103–123, doi:10.1016/S0341-8162(01)00161-8, 2002.
- Rose, L. A., D. L. Karwan, and S. E. Godsey, Concentration–discharge Relationships Describe Solute and Sediment Mobilization, Reaction, and Transport at Event and Longer Timescales, *Hydrological Processes*, 32(18), 2829–2844, doi:10.1002/hyp.13235, 2018.
- Rossi, L., N. Chèvre, R. Fankhauser, J. Margot, R. Curdy, M. Babut, and D. A. Barry, Sediment Contamination Assessment in Urban Areas Based on Total Suspended Solids, *Water Research*, 47(1), 339–350, doi:10.1016/j.watres.2012.10.011, 2013.
- Royer, J. R., E. I. Corwin, A. Flior, M.-L. Cordero, M. L. Rivers, P. J. Eng, and H. M. Jaeger, Formation of Granular Jets Observed by High-Speed X-Ray Radiography, *Nature Physics*, 1, 164–167, doi:10.1038/nphys175, 2005.
- Ryżak, M., A. Bieganski, and C. Polakowski, Effect of Soil Moisture Content on the Splash Phenomenon Reproducibility, *PLOS ONE*, 10(3), e0119269, doi:10.1371/journal.pone.0119269, 2015.
- Sadeghi, S. H., M. Kiani Harchegani, and H. Asadi, Variability of Particle Size Distributions of Upward/Downward Splashed Materials in Different Rainfall Intensities and Slopes, *Geoderma*, 290, 100–106, doi:10.1016/j.geoderma.2016.12.007, 2017a.
- Sadeghi, S. H., M. Zabihi, M. Vafakhah, and Z. Hazbavi, Spatiotemporal Mapping of Rainfall Erosivity Index for Different Return Periods in Iran, *Natural Hazards*, 87(1), 35–56, doi:10.1007/s11069-017-2752-3, 2017b.
- Sadeghi, S. H. R., T. Mizuyama, S. Miyata, T. Gomi, K. Kosugi, T. Fukushima, S. Mizugaki, and Y. Onda, Determinant Factors of Sediment Graphs and Rating Loops in a Reforested Watershed, *Journal of Hydrology*, 356(3-4), 271–282, doi:10.1016/j.jhydrol.2008.04.005, 2008.

Bibliography

- Salant, N. L., M. A. Hassan, and C. V. Alonso, Suspended Sediment Dynamics at High and Low Storm Flows in Two Small Watersheds, *Hydrological Processes*, 22(11), 1573–1587, doi:10.1002/hyp.6743, 2008.
- Sander, G. C., P. B. Hairsine, C. W. Rose, D. Cassidy, J.-Y. Y. Parlange, W. L. Hogarth, and I. G. Lisle, Unsteady Soil Erosion Model, Analytical Solutions and Comparison with Experimental Results, *Journal of Hydrology*, 178(1-4), 351–367, doi:10.1016/0022-1694(95)02810-2, 1996.
- Sander, G. C., J. Y. Parlange, D. A. Barry, M. B. Parlange, and W. L. Hogarth, Limitation of the Transport Capacity Approach in Sediment Transport Modeling, *Water Resources Research*, 43(2), doi:10.1029/2006WR005177, 2007.
- Sander, G. C., T. Zheng, P. Heng, Y. Zhong, and D. A. Barry, Sustainable Soil and Water Resources: Modelling Soil Erosion and Its Impact on the Environment, in *19th International Congress on Modelling and Simulation*, pp. 45–56, Modelling and Simulation Society of Australia and New Zealand Inc., Perth, Australia, <http://mssanz.org.au/modsim2011/plenary.htm>, 2011.
- Sassolas-Serrayet, T., R. Cattin, and M. Ferry, The Shape of Watersheds, *Nature Communications*, 9(1), 3791, doi:10.1038/s41467-018-06210-4, 2018.
- Schumm, S. A., and H. R. Khan, Experimental Study of Channel Patterns, *Nature*, 233(5319), 407–409, doi:10.1038/233407a0, 1971.
- Seeger, M., M.-P. Errea, S. Beguería, J. Arnáez, C. Martí, and J. García-Ruiz, Catchment Soil Moisture and Rainfall Characteristics as Determinant Factors for Discharge/Suspended Sediment Hysteretic Loops in a Small Headwater Catchment in the Spanish Pyrenees, *Journal of Hydrology*, 288(3-4), 299–311, doi:10.1016/j.jhydrol.2003.10.012, 2004.
- Seidl, M., and W. Dietrich, The Problem of Channel Erosion into Bedrock, *Catena*, 23(Supplement), 101–104, 1992.
- Shen, H., F. Zheng, L. Wen, J. Lu, and Y. Jiang, An Experimental Study of Rill Erosion and Morphology, *Geomorphology*, 231, 193–201, doi:10.1016/j.geomorph.2014.11.029, 2015.

- Sheng, M., K. Xiong, L. Wang, X. Li, R. Li, and X. Tian, Response of Soil Physical and Chemical Properties to Rocky Desertification Succession in South China Karst, *Carbonates and Evaporites*, 33(1), 15–28, doi:10.1007/s13146-016-0295-4, 2018.
- Sherriff, S. C., J. S. Rowan, O. Fenton, P. Jordan, A. R. Melland, P.-E. E. Mellander, and D. O. Huallacháin, Storm Event Suspended Sediment-Discharge Hysteresis and Controls in Agricultural Watersheds: Implications for Watershed Scale Sediment Management, *Environmental Science and Technology*, 50(4), 1769–1778, doi:10.1021/acs.est.5b04573, 2016.
- Shit, P. K., G. S. Bhunia, and R. Maiti, Developments of Rill Networks: An Experimental Plot Scale Study, *Journal of Water Resource and Protection*, 05(02), 133, doi:10.4236/jwarp.2013.52015, 2013.
- Shreve, R. L., Infinite Topologically Random Channel Networks, *The Journal of Geology*, 75(2), 178–186, doi:https://doi.org/10.1086/627245https://doi.org/10.1086/627245, 1967.
- Shreve, R. L., Stream Lengths and Basin Areas in Topologically Random Channel Networks, *The Journal of Geology*, 77(4), 397–414, doi:https://doi.org/10.1086/628366, 1969.
- Shumilovskikh, L. S., et al., Landscape Evolution and Agro-Sylvo-Pastoral Activities on the Gorgan Plain (NE Iran) in the Last 6000 Years, *The Holocene*, 26(10), 1676–1691, doi:10.1177/0959683616646841, 2016.
- Simpson, G., and S. Castelltort, Coupled model of surface water flow, sediment transport and morphological evolution, *Computers & Geosciences*, 32(10), 1600–1614, doi:10.1016/j.cageo.2006.02.020, 2006.
- Simpson, G., and F. Schlunegger, Topographic evolution and morphology of surfaces evolving in response to coupled fluvial and hillslope sediment transport, *108*, 1–16, doi:10.1029/2002JB002162, 2003.
- Sinclair, H., Earth Science: Making a Mountain out of a Plateau, *Nature*, 542(7639), 36–41, doi:10.1038/542041a, 2017.
- Singh, A., L. Reinhardt, and E. Foufoula-Georgiou, Landscape Reorganization under Changing Climatic Forcing: Results from an Experimental Landscape, *Water Resources Research*, 51(6), 4320–4337, doi:10.1002/2015WR017161, 2015.

Bibliography

- Sklar, L. S., W. E. Dietrich, D. W. E. Sklar, L. S. Sklar, and W. E. Dietrich, Sediment and rock strength controls on river incision into bedrock, *Geology*, 29(12), 1087, doi:10.1130/0091-7613(2001)029<1087:SARSCO>2.0.CO;2, 2001.
- Slattery, M. C., and R. B. Bryan, Hydraulic Conditions for Rill Incision under Simulated Rainfall: A Laboratory Experiment, *Earth Surface Processes and Landforms*, 17(2), 127–146, doi:10.1002/esp.3290170203, 1992.
- Smith, H. G., and D. Dragovich, Interpreting Sediment Delivery Processes Using Suspended Sediment-Discharge Hysteresis Patterns from Nested Upland Catchments, South-Eastern Australia, *Hydrological Processes*, 23(17), 2415–2426, doi:10.1002/hyp.7357, 2009.
- Smith, R., D. Goodrich, D. Woolhiser, and C. Unkrich, KINEROS-a Kinematic Runoff and Erosion Model, in *Computer Models of Watershed Hydrology*, pp. 697–732, Highlands Ranch, Colorado: Water Resources Publications, 1995.
- Smith, T. R., Analytic Theory of Equilibrium Fluvial Landscapes: The Integration of Hillslopes and Channels, *Journal of Geophysical Research: Earth Surface*, 123(3), 557–615, doi:10.1002/2016JF004073, 2018.
- Smith, T. R., and F. P. Bretherton, Stability and the Conservation of Mass in Drainage Basin Evolution, *Water Resources Research*, 8(6), 1506, doi:10.1029/WR008i006p01506, 1972.
- Stark, C. P., An Invasion Percolation Model of Drainage Network Evolution, *Nature*, 352(6334), 423–425, doi:10.1038/352423a0, 1991.
- Stefanon, L., L. Carniello, A. D'Alpaos, and A. Rinaldo, Signatures of sea level changes on tidal geomorphology: Experiments on network incision and retreat, *Geophysical Research Letters*, 39(12), 1–6, doi:10.1029/2012GL051953, 2012.
- Stokes, G. G., On the Effect of the Internal Friction of Fluids on the Motion of Pendulums, in *Transactions of the Cambridge Philosophical Society*, vol. 9, p. 8, Cambridge University Press, Cambridge, doi:10.1017/CBO9780511702242.005, 1851.
- Strohmeier, S., G. Laaha, H. Holzmann, and A. Klik, Magnitude and Occurrence Probability of Soil Loss: A Risk Analytical Approach for the Plot Scale for Two Sites

- in Lower Austria, *Land Degradation and Development*, 27(1), 43–51, doi:10.1002/ldr.2354, 2016.
- Sun, L., M. Yan, Q. Cai, and H. Fang, Suspended Sediment Dynamics at Different Time Scales in the Loushui River, South-Central China, *Catena*, 136, 152–161, doi:10.1016/j.catena.2015.02.014, 2016.
- Sweeney, K. E., J. J. Roering, and C. Ellis, Experimental Evidence for Hillslope Control of Landscape Scale, *Science*, 349(6243), 51–53, doi:10.1126/science.aab0017, 2015.
- Tarboton, D. G., A new method for the determination of flow directions and upslope areas in grid digital elevation models, *Water Resources Research*, 33(2), 309–319, doi:10.1029/96WR03137, 1997.
- Tarboton, D. G., R. L. Bras, and I. Rodriguez-Iturbe, The Fractal Nature of River Networks, *Water Resources Research*, 24(8), 1317–1322, doi:10.1029/WR024i008p01317, 1988.
- Tarboton, D. G., R. L. Bras, and I. Rodríguez-Iturbe, Scaling and Elevation in River Networks, *Water Resources Research*, 25(9), 2037–2051, doi:10.1029/WR025i009p02037, 1989.
- Tatard, L., O. Planchon, J. Wainwright, G. Nord, D. Favis Mortlock, N. Silvera, O. Ribolzi, M. Esteves, and C. H. Huang, Measurement and Modelling of High-Resolution Flow-Velocity Data under Simulated Rainfall on a Low-Slope Sandy Soil, *Journal of Hydrology*, 348(1-2), 1–12, doi:10.1016/j.jhydrol.2007.07.016, 2008.
- Taylor, P. J., and R. Leigh, An Integral Approach to Bedrock River Profile Analysis, *Earth Surface Processes and Landforms*, 38(6), 570–576, doi:10.1002/esp.3302, 2013.
- Theodoratos, N., H. Seybold, and J. W. Kirchner, Scaling and Similarity of a Stream-Power Incision and Linear Diffusion Landscape Evolution Model, *Earth Surface Dynamics*, 6(3), 779–808, doi:https://doi.org/10.5194/esurf-6-779-2018, 2018.
- Tilman, D., K. G. Cassman, P. A. Matson, R. Naylor, and S. Polasky, Agricultural Sustainability and Intensive Production Practices, *Nature*, doi:10.1038/nature01014, 2002.

Bibliography

- Tromp-van Meerveld, H. J., J.-Y. Parlange, D. A. Barry, M. F. Tromp, G. C. Sander, M. T. Walter, and M. B. Parlange, Influence of Sediment Settling Velocity on Mechanistic Soil Erosion Modeling, *Water Resources Research*, 44(6), doi:10.1029/2007WR006361, 2008.
- Tucker, G. E., and G. R. Hancock, Modelling landscape evolution, *Earth Surface Processes and Landforms*, 35(1), 28–50, doi:10.1002/esp.1952, 2010.
- Tucker, G. E., and R. L. Slingerland, Erosional Dynamics, Flexural Isostasy, and Long-Lived Escarpments: A Numerical Modeling Study, *Journal of Geophysical Research: Solid Earth*, 99(B6), 12,229–12,243, doi:10.1029/94JB00320, 1994.
- Turowski, J. M., D. Lague, A. Crave, and N. Hovius, Experimental Channel Response to Tectonic Uplift, *Journal of Geophysical Research: Earth Surface*, 111(F3), F03,008, doi:10.1029/2005JF000306, 2006.
- van der Meij, W. M., A. J. A. M. Temme, H. S. Lin, H. H. Gerke, and M. Sommer, On the Role of Hydrologic Processes in Soil and Landscape Evolution Modeling: Concepts, Complications and Partial Solutions, *Earth-Science Reviews*, doi:10.1016/j.earscirev.2018.09.001, 2018.
- Van Oost, K., L. Beuselinck, P. B. Hairsine, and G. Govers, Spatial Evaluation of a Multi-Class Sediment Transport and Deposition Model, *Earth Surface Processes and Landforms*, 29(8), 1027–1044, doi:10.1002/esp.1089, 2004.
- Vanmaercke, M., F. Ardizzone, M. Rossi, and F. Guzzetti, Exploring the Effects of Seismicity on Landslides and Catchment Sediment Yield: An Italian Case Study, *Geomorphology*, 278, 171–183, doi:10.1016/j.geomorph.2016.11.010, 2017.
- Viani, J.-P., Contribution à l'étude Expérimentale de l'érosion Hydrique, PhD Thesis, Ph.D. Thesis, Ecole Polytech. Féd. de Lausanne (EPFL), Lausanne, Switzerland, doi:10.5075/epfl-thesis-641, doi:10.5075/epfl-thesis-641, 1986.
- Walker, J. D., M. T. Walter, J.-Y. Parlange, C. W. Rose, H. J. T.-v. Meerveld, B. Gao, and A. M. Cohen, Reduced Raindrop-Impact Driven Soil Erosion by Infiltration, *Journal of Hydrology*, 342(3), 331–335, doi:10.1016/j.jhydrol.2007.06.003, 2007.
- Walling, D. E., and B. W. Webb, Estimating the Discharge of Contaminants to Coastal Waters by Rivers: Some Cautionary Comments, *Marine Pollution Bulletin*, 16(12), 488–492, doi:10.1016/0025-326X(85)90382-0, 1985.

- Wang, L., Z. Shi, J. Wang, N. Fang, G. Wu, and H. Zhang, Rainfall Kinetic Energy Controlling Erosion Processes and Sediment Sorting on Steep Hillslopes: A Case Study of Clay Loam Soil from the Loess Plateau, China, *Journal of Hydrology*, 512, 168–176, doi:10.1016/j.jhydrol.2014.02.066, 2014.
- Watkins, N. W., G. Pruessner, S. C. Chapman, N. B. Crosby, and H. J. Jensen, 25 Years of Self-Organized Criticality: Concepts and Controversies, *Space Science Reviews*, 198(1-4), 3–44, doi:10.1007/s11214-015-0155-x, 2016.
- Whipple, K. X., A. M. Forte, R. A. DiBiase, N. M. Gasparini, and W. B. Ouimet, Timescales of landscape response to divide migration and drainage capture: Implications for the role of divide mobility in landscape evolution, *Journal of Geophysical Research: Earth Surface*, doi:10.1002/2016JF003973, 2016.
- Whipple, K. X., R. A. DiBiase, W. B. Ouimet, and A. M. Forte, Preservation or piracy: Diagnosing low-relief, high-elevation surface formation mechanisms, *Geology*, 45(1), 91–94, doi:10.1130/G38490.1, 2017.
- Willemin, J. H., Hack's Law: Sinuosity, Convexity, Elongation, *Water Resources Research*, 36(11), 3365–3374, doi:10.1029/2000WR900229, 2000.
- Willett, S. D., S. W. McCoy, J. T. Perron, L. Goren, and C. Y. Chen, Dynamic reorganization of river basins, *Science*, 343(6175), 1248,765, doi:10.1126/science.1248765, 2014.
- Willgoose, G., R. L. Bras, and I. Rodriguez-Iturbe, A Coupled Channel Network Growth and Hillslope Evolution Model: 2. Nondimensionalization and Applications, *Water Resources Research*, 27(7), 1685–1696, doi:10.1029/91WR00936, 1991.
- Willgoose, G., R. L. Bras, and I. Rodriguez-Iturbe, The Relationship between Catchment and Hillslope Properties: Implications of a Catchment Evolution Model, *Geomorphology*, 5(1-2), 21–37, doi:10.1016/0169-555X(92)90056-T, 1992.
- Willgoose, G. R., A Physically Based Channel Network and Catchment Evolution Model, PhD Thesis, Massachusetts Institute of Technology, 1989.
- Williams, G. P., Sediment Concentration versus Water Discharge during Single Hydrologic Events in Rivers, *Journal of Hydrology*, 111(1-4), 89–106, doi:10.1016/0022-1694(89)90254-0, 1989.

Bibliography

- Wittmann, R., T. Kautzky, A. Hübler, and E. Lüscher, A simple experiment for the examination of dendritic river systems, *Naturwissenschaften*, 78(1), 23–25, doi:10.1007/BF01134037, 1991.
- Woldenberg, M. J., HORTON'S LAWS JUSTIFIED IN TERMS OF ALLOMETRIC GROWTH AND STEADY STATE IN OPEN SYSTEMS, *GSA Bulletin*, 77(4), 431–434, doi:10.1130/0016-7606(1966)77[431:HLJITO]2.0.CO;2, 1966.
- Wood, P. J., and P. D. Armitage, Biological Effects of Fine Sediment in the Lotic Environment, *Environmental Management*, 21(2), 203–217, doi:10.1007/s002679900019, 1997.
- Wu, X., Y. Wei, J. Wang, J. Xia, C. Cai, and Z. Wei, Effects of Soil Type and Rainfall Intensity on Sheet Erosion Processes and Sediment Characteristics along the Climatic Gradient in Central-South China, *Science of The Total Environment*, 621, 54–66, doi:10.1016/j.scitotenv.2017.11.202, 2018.
- Yang, D., S. Kanae, T. Oki, T. Koike, and K. Musiak, Global Potential Soil Erosion with Reference to Land Use and Climate Changes, *Hydrological Processes*, 17(14), 2913–2928, doi:10.1002/hyp.1441, 2003.
- Yang, R., S. D. Willett, and L. Goren, In situ low-relief landscape formation as a result of river network disruption, *Nature*, 520(7548), 526–9, doi:10.1038/nature14354, 2015.
- Yao, C., T. Lei, W. J. Elliot, D. K. McCool, J. Zhao, and S. Chen, Critical Conditions for Rill Initiation, *Transactions of the ASABE*, 51(1), 107–114, 2008.
- Yeshaneh, E., A. Eder, and G. Blöschl, Temporal Variation of Suspended Sediment Transport in the Koga Catchment, North Western Ethiopia and Environmental Implications, *Hydrological Processes*, 28(24), 5972–5984, doi:10.1002/hyp.10090, 2014.
- Zare, M., T. Panagopoulos, and L. Loures, Simulating the Impacts of Future Land Use Change on Soil Erosion in the Kasilian Watershed, Iran, *Land Use Policy*, 67, 558–572, doi:10.1016/j.landusepol.2017.06.028, 2017.
- Zhang, Q., M. Gao, R. Zhao, and X. Cheng, Scaling of Liquid-Drop Impact Craters in Wet Granular Media, *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 92(4), 1–8, doi:10.1103/PhysRevE.92.042205, 2015.

- Zhao, S.-C., R. de Jong, and D. van der Meer, Raindrop Impact on Sand: A Dynamic Explanation of Crater Morphologies, *Soft Matter*, 11(33), 6529–6718, doi:10.1039/C5SM00957J, 2015.
- Zhong, Y., Modelling Sediment Transportation and Overland Flow, PhD Thesis, Ph.D. thesis, University of Oxford, Oxford, United Kingdom, <https://ora.ox.ac.uk:443/objects/uuid:a45eefae-5a0f-4917-9abb-261ae792f2ee>, 2013.
- Zhou, J., B. Fu, G. Gao, Y. Lü, Y. Liu, N. Lü, and S. Wang, Effects of Precipitation and Restoration Vegetation on Soil Erosion in a Semi-Arid Environment in the Loess Plateau, China, *Catena*, 137, 1–11, doi:10.1016/j.catena.2015.08.015, 2016.
- Zhou, P., O. Luukkanen, T. Tokola, and J. Nieminen, Effect of Vegetation Cover on Soil Erosion in a Mountainous Watershed, *Catena*, 75(3), 319–325, doi:10.1016/j.catena.2008.07.010, 2008.



Mohsen Cheraghi

Curriculum Vitae

Education

- 2013–2018 **PhD in Civil and Environmental Engineering**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland.
Thesis title: Stochastic and physical characterization of overland flow morphologies
- 2010–2013 **Master of Science in Mechanical Engineering (Energy conversion)**, *University of Tehran*, Iran, *GPA – 17.83/20*.
Thesis title: Mixing and heat transfer enhancement in channels via a rotationally oscillating cylinder (20/20)
- 2006–2010 **Bachelor of Science in Mechanical Engineering**, *University of Tehran*, Iran, *GPA – 17.38/20*.
Thesis title: Investigation of fluid flow and heat transfer enhancement in a mini-channel via a circular cylinder at different heights (20/20)

Scientific and technical expertise

I use **Python** and **C++** for data processing, visualization, scripting and numerical simulations. I am an expert of **Numerical Simulations**, **Data Analysis** and **Optimization**, so far working on the engineering designs and physical processes such as energy, heat transfer, fluid mechanics and sediment transport. My full publication list can be found at: <https://scholar.google.ch/citations?user=QxQsP5MAAAAJ&hl=en>

Peer-reviewed journal publication

M. Cheraghi, A. Rinaldo, G. C. Sander, P. Perona, D. A. Barry (2018), Catchment drainage network scaling laws found experimentally in overland flow morphologies, *Geophysical Research Letters* 45, Accepted.

I. Lisle, G. C. Sander, J-Y. Parlange, CW. Rose et al. (2017), Transport time scales in soil erosion modeling, *Vadose Zone Journal*, 16.

M. Cheraghi, S. Jomaa, G. C. Sander, and D. A. Barry (2016), Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects, *Water Resources Researches*, 52, 8613-8629.

Avenue de Valmont 16 – 1010 Lausanne Switzerland

☎ (+41)78 700 44 04 • ☎ (+41)21 693 59 19

✉ mohsen.cheraghi@epfl.ch, msn.cheraghi@gmail.com

🌐 <https://people.epfl.ch/mohsen.cheraghi>

M. Cheraghi, M. Raisee and M. Moghaddami (2014), Effect of cylinder proximity to the wall on channel flow heat transfer enhancement, *Comptes Rendus Mécanique*, 342, 63-72.

A. Beskok, M. Raisee, B. Celik, B. Yagiz and M. Cheraghi (2012), Heat transfer enhancement in a channel via rotationally oscillating cylinder, *Int. J. Thermal Sciences*, 58, 61-69.

M. Raisee & M. Cheraghi (2012), Gas mixing enhancement in mini-channels using a rotationally oscillatory circular cylinder, *Journal of Physics Conference Series* 362(1), 012-041.

Work experiences

- 2010–2013 In parallel with my master studies, I worked with an engineering team (MAPNA group, Iran) as a C++ programmer for the calculations and data analysis. A software was produced for the optimum gas turbine cooling system at different climates (economics, water consumption, applicability)
- Summer 2010 Internship in Tehran Refinery, Iran
- 2006-2008 Working as a consultant in a high school

work-related experiences

- Workshops and conferences Machine Learning for Environmental and Geosciences [Zürich], American Geophysical Union [San Francisco], European Geophysical Union [Vienna], Transport & Sedimentation of Solid particles [Delft], OpenFOAM workshop [Zagreb]
- Field measurement During my PhD, I had a life-changing 3-month stay in China (Nanjing, Hohai university) for field measurement to capture a high resolution Digital Elevation Model of a marshland
- Selected Courses DATA analysis, Optimization, Turbulence, Particle based methods, CFD-DEM workshop (Austria), PEST workshop (Neuchâtel), Heat transfer, Computational fluid dynamics
- Teaching Modflow software for the “Underground water”, Introduction to C++, Fluid mechanics

Computer skills

- Programming Python, Tensorflow, C++, Fortran, MATLAB
- Application OpenFOAM, FLUENT, Solidworks, PEST, EDEM, MODFLOW, Thermoflow

Honorable Mentions

- 2013 PhD admission at EPFL, Switzerland
- 2012 Awarded in recognition of Scientific Elite, National Elite Institute, Iran
- 2012 Distinguished student as the Exceptional Talent between M.S students in the University of Tehran
- 2010 Awarded full scholarship from University of Tehran for M.S degree

Avenue de Valmont 16 – 1010 Lausanne Switzerland

☎ (+41)78 700 44 04 • ☎ (+41)21 693 59 19

✉ mohsen.cheraghi@epfl.ch, msn.cheraghi@gmail.com

🌐 <https://people.epfl.ch/mohsen.cheraghi>

- 2010 Distinguished student as the Exceptional Talent between B.S students in the University of Tehran
- 2010 Ranked 416th in Graduate University Entrance Exam among more than 400,000 participants, Tehran, Iran
- summer 2006 Semifinalist in exam of Iran's national Physics Olympiad, Tehran, Iran
- summer 2006 Semifinalist in exam of Iran's national Chemistry Olympiad, Tehran, Iran

Other information

- Language I am a **Persian** native speaker, and I am fluent in spoken and written **English**. I have an above-average command **French (B2)**, and a very basic knowledge of **German**
- Interests I am an amateur harmonica player and very interested in poet and literature. My main sports are fitness, badminton, swimming, and sometimes skiing
- Social Activities I involved in social activities like: Representative of PhD students (EPFL) (2015-16), Event organiser in Iranian Student association at Lausanne (IRSA) (2015-16), Fundraising for an earthquake in Kermanshah, Iran (2018)
- Nationality Iranian, born in 20 May 1988
- References Available under request

