1. Introduction

The interplay of a variety of physical processes acting over a range of space and time scales is responsible for the evolution of Earth’s surface topography. These physical processes broadly fall in four categories: (i) river incision that leads to advective transport of material over distances of up to several thousands of km, (ii) diffusive processes such as soil creep and landslides that operate over comparatively shorter length scales, (iii) large-scale surface deformation arising from tectonic forces in the crust and lithosphere, and (iv) transient long-wavelength but small-amplitude topographic variations arising from deep mantle convective forces that operate over timescales of tens of millions of years [1]. In recent decades, numerical models of geomorphological evolution have become an important tool to facilitate a better understanding of these coupled processes that sculpt surface topography [2–9].

Geomorphological (or landscape) evolution models have made rapid advances in recent decades as a result of the increasing computational capacity of computing hardware. However, the numerical resolution necessary to study geomorphological evolution over continental scales, spanning tens of millions of years – particularly aimed at better understanding the influence of long wavelength but small amplitude dynamic topography – still poses significant computational challenges. Wallis et al. [10] and Do et al. [11] presented parallel flow routing algorithms based on the message-passing-interface (MPI) on clusters and implementations on graphics processing units (GPUs) have also been reported [12,13], which
show significant speedups. Braun and Willett [9] presented a parallelization approach based on open-multi-processing (OpenMP), producing near-linear scalability on a shared-memory machine. However, both MPI- and GPU-based parallelization strategies involve increased development and debugging times and are likely to compromise ease of code adaptability. We therefore adopt the parallelization approach described in Braun and Willett [9] for its simplicity and ease of implementation.

2. Software framework

SPGM has been implemented using C++ [14] and the basic abstractions utilized are shown in Fig. 1. The Config class is responsible for parsing input configuration files and for providing collaborating classes access to named parameters and parameter-groups. Each of the physical processes implemented in SPGM – which we interchangeably refer to as 'modules', e.g. Precipitation – inherit from the abstract Process class. The Process class contains a reference to the Config class and its sub-classes (blue rectangle in Fig. 1) implement the Execute() function, which contains the core numerical algorithm pertaining to a given physical process. The ModelBuilder class instantiates a number of 'Processes' based on an input configuration file and populates an instance of the Model class with them. The SurfaceTopology class manages the underlying geometry of the computational mesh and recomputes drainage networks at the beginning of each time step, as shown in the program flowchart in Fig. 2.

In the following sections we describe the parsing of configuration files where general parameter values, as well as time-series data for specific parameters can be specified. We also briefly describe the algorithms used in mesh generation, flow-routing and output generation.

2.1. Configuration files and parameter time series

We use simple, plain-text configuration files to list a number of required parameters, followed by several groups of parameters corresponding to physical processes modeled. Parameter values
Fig. 2. Program flowchart of SPGM. See Fig. 1 for the relationship between Process and Model classes.

such as rate of precipitation or uplift can be specified in the configuration file as either: (i) a scalar value, (ii) a time-series in a text file containing a sequence of (time, value) pairs, or (iii) a time-series in a text file containing a sequence of (time, file) pairs—where each file lists nodal values at that time, thus allowing space-time variability (see Supplementary Data). Linear interpolation is used in the second and third options for model times that fall in between times at which parameter values are provided.

2.2. The computational mesh

The initial topography (xyz triplets) and boundary conditions (BC) can be specified in a simple four-columned text file. Only arbitrary Dirichlet boundary conditions are currently supported and nodes with a fixed elevation are marked by ‘1’ in the fourth (BC) column—free nodes are marked by ‘0’. Output from a given model can also be used as the initial topography of another model (see Appendix C.2 for more details).

We compute a Delaunay triangulation and the corresponding Voronoi tessellation of the initial point-data on the x-y plane using the divide and conquer method [15], following the algorithm outlined in Lischinski [16]. Numerical solutions obtained in the various modules, described in latter sections, directly update the height-field on this computational mesh. Each point in the triangulation is connected to a set of neighbors – also known as ‘natural neighbors’ – and its associated Voronoi polygon marks the drainage area associated with the point. For highly irregular distributions of initial point-data, iterative Laplacian smoothing
can be applied as:

\[ x_i = \phi \frac{1}{N} \sum_{j=1}^{N} x_j, \]  

(1)

where, at each iteration, \( \phi \) is the smoothing factor, \( N \) is the number of neighboring vertices to node \( i \), \( x_i \) is the position of the \( j \)th node of the Voronoi polygon corresponding to the \( i \)th node, rate of precipitation that can vary in space and time and the time-step in years, respectively.

3.2. Uplift

This module incorporates uplift (tectonic and dynamic being indistinguishable) based on uplift rate, \( U \), which can be specified as a constant or as a time-series—see Section 2.1. The latter option allows for variable uplift rates in both space and time.

3.3. Detachment-limited fluvial transport

In a detachment-limited scenario, fluvial incision can be described by a power law function [20]:

\[ \frac{\partial z}{\partial t} = -K_f A^n (\nabla z)^n, \]  

(3)

where \( z \), \( t \), \( K_f \) and \( A \) are elevation, time, erodability constant and contributing drainage area, respectively. The value of \( K_f \) depends on a number of factors, including lithology and climate, while its dimension depends on dimensionless constants \( m \) and \( n \) (that both have small positive values in which their ratio, \( m/n \), is generally \( \approx 0.5 \) [20].

The solution to this equation is obtained from the implicit scheme described in Braun and Willett [9] that ensures numerical stability even when large time-steps are used—although, larger time-steps may result in lower accuracy (see Fig. 4d in Braun and Willett [9]. The solution scheme traverses the nodes according to \( S \), thus following the donor-link to incrementally progress upstream, starting from outlet nodes (specified as BCs), which have a fixed elevation. When \( n = 1 \) in Eq. (3), it can be solved explicitly to compute the evolving elevation of a node. Otherwise, when \( n \neq 1 \) in Eq. (3), an iterative Newton–Raphson scheme [21] is used—see Braun and Willett [9] for a detailed derivation of the approach. The solution scheme is parallelized using OpenMP over the total number of catchments identified, since nodes that belong to a given catchment can be processed in isolation from nodes in other catchments—see Section 5 for scalability results.

We test this module with a simple model, where the initial topography of a 100 km x 100 km region is set to a uniform elevation of 500 m. Boundary conditions are set on nodes along \( y = 0 \), with their elevation set to 0 and an uplift function is imposed from time \( t = 0 \) as:

\[ U = U_0 \frac{y}{y_{max}}. \]  

(4)

The model is computed with \( m = 0.5, n = 1, K_f = 6 \times 10^{-4} \) yr\(^{-1} \), \( \Delta t = 100 \) yrs and \( U_0 = 5 \times 10^{-3} \) m yr\(^{-1} \) (see Supplementary Data). These parameters were chosen such that topographic evolution (Fig. 3A) reaches a steady-state at \( \approx 3 \times 10^4 \) years (Fig. 3B) and visually resembles results shown in Fig. 4 of Braun and Willett [9].

3.4. Transport-limited fluvial transport

We traverse nodes according to \( S \) in reverse order (see Section 2.3) to incrementally compute volumetric discharge, \( D \), through each node. We implement the mathematical model in Kooi and Beaumont [22], where local sediment flux, \( Q \), along river networks can potentially be at a disequilibrium with carrying capacity, \( Q_c \), which is defined as:

\[ Q_c = K_f D^n (\nabla z)^n, \]  

(5)

where \( m \) and \( n \) are dimensionless constants that vary around 1—see Tucker and Slingerland [23] for a detailed derivation of \( Q_c \).
Fig. 3. Model topography at 100,000 years is shown in (A) and evolution of mean topography through time is shown in (B).

Fig. 4. Initial model topography is shown in (A). Model topography at $20 \times 10^6$ years is shown in (B), with positive changes in elevation, resulting from deposition of sediments, contoured at 45 m intervals to highlight the alluvial fans at the foothills. Evolution of mean topography through time is shown in (C).

River networks evolve towards equilibrium at a rate proportional to the disequilibrium present [22]. Sediments are deposited when $Q > Q_e$ and changes in elevation are given by:

$$\frac{\partial z}{\partial t} = \left( Q - Q_e \right) a_i.$$  

(6)

Material becomes entrained when $Q < Q_e$ and changes in elevation are given by:

$$\frac{\partial z}{\partial t} = l_i \left( Q - Q_e \right),$$  

(7)

where $l_i$ and $l_s$ are the local channel length and the erosion length-scale, respectively. Local sediment flux, $Q$, thus evolves towards an equilibrium with the carrying capacity, eroding and depositing material in the process.

We assume $l_i$ to be larger for bedrock, compared to that for previously deposited alluvial sediments—see Appendix A for more details. We also record sediment accumulation throughout model evolution so that appropriate values of $l_i$ are used for fluvial entrainment at a given location, depending on the presence of previously deposited sediments. We traverse nodes according to $S$ in reverse order and solve Eqs. (6) and (7) explicitly, based on simple continuity of mass. This solution scheme is equally amenable to parallelization as that in Section 3.3 and scalability results are discussed in Section 5.

We test this module with a simple model, where the initial topography in a 1000 km $\times$ 1000 km region is derived based on the complementary error function, with a small amount Perlin noise [24] added to induce the development of realistic river networks. Boundary conditions are set on nodes along $y = 0$, with their elevation set to 0. The model is computed with parameters chosen arbitrarily to demonstrate a simple case of topographic evolution ($m = 1, n = 0.5, K_f = 5 \times 10^{-2}, \Delta t = 1000$ yrs and $\lambda = 0.2$ m yr$^{-1}$); these parameters result in a realistic landscape and sediment distribution after $20 \times 10^6$ years (Fig. 4A), with a reasonable evolution of mean topography over this period (Fig. 4B).

Note that since sediment transport and deposition in a detachment-limited scenario depends on the surface runoff through each node, this module must appear in the parameter file (see Supplementary Data) after the precipitation module.

3.5. Short-range hill-slope processes

Short-range material transport due to processes e.g. soil creep, landslides, etc. is approximated by anisotropic diffusion:

$$\frac{\partial z}{\partial t} = \nabla \cdot \left( K_d(x, y, t) \nabla z \right).$$  

(8)

where $K_d$ is a space-time varying diffusivity field. Diffusivities for both bedrock and sediments are specified in parameter files.
Fig. 5. Model topography and diffusivity at $20 \times 10^6$ years are shown in (A) and (B), respectively. Positive changes in elevation resulting from deposition of sediments are contoured at 70 m intervals to highlight the alluvial fans at the foothills in (A). Evolution of mean topography through time is shown in (C).

Fig. 6. (A) Speedups in the detachment-limited module, based on an average of 10 time-steps, are shown as a function of the number of cores for a fixed problem size (i.e. strong scaling), for a range of grid densities. The light dashed line shows perfect speedup. (B) The same as in (A), but for the transport-limited module.

(see Appendix A for more details). We assume an absence of any sediments at the start of the model—consequently, as material gets entrained and sediments are routed throughout model evolution, the distribution of diffusivities vary in space–time, as described in Section 3.4. We cast Eq. (8) as a simple finite element problem [25], based on $P_1$ elements on the triangulated mesh. We use an implicit Euler time integration scheme and solve the resulting sparse system of equations using the conjugate gradient iterative solver implemented in the Eigen library [26]. A detailed verification procedure for accessing the numerical implementation is given in Appendix B.

To further demonstrate the effects of this module, we impose it on the simple model in Section 3.4—see Appendix A for descriptions of additional parameters pertaining to this module. Model results are shown in Fig. 5, where the spatial distribution of diffusivities (Fig. 5B) reflect sediments deposited at the foothills.

4. Illustrative examples

In Appendix C we describe two models—the first shows the effect of transient dynamic topography on the evolution and reorganization of drainage networks; the second shows the influence of the evolution of drainage networks when river systems transition from a detachment-limited to a transport-limited state. These models are based on the simple model described in Section 3.5 and serve to demonstrate how the various modules can be brought together to explore a broad range of geomorphological scenarios. Annotated parameter-files and initial topographies for models used to produce Figs. C.9 and C.10 can be found online in Supplementary Data.

5. Discussion

In order to demonstrate the parallel scalability of the fluvial transport modules, we have computed models with a similar initial topography as that in Appendix C.1, but with a range of grid densities. Fig. 6A and 6B show strong scaling results for the detachment-limited and transport-limited modules, respectively. The scaling tests were computed on a desktop machine with two Intel Xeon E5-2650 processors, each with eight cores (2.6 GHz clock-speed). The parallelized solution schemes in both the detachment-limited and the transport-limited modules show near-linear scalability for the densest mesh ($1600 \times 1600$), while scalability gradually drops off with increasingly smaller mesh sizes in both. Our results suggest that an order of magnitude speedup can be achieved on typical desktop machines—thus making continental scale models with tens of millions of nodes more computationally feasible.

Studying continental scale geomorphological evolution over tens of millions of years, particularly in order to better understand the influences of long wavelength but small amplitude dynamic topography, has been computationally prohibitive. Moreover, since paleotopography is poorly constrained over geological timescales, models of geomorphological evolution generally start with synthetic initial topography conditions—thus, the ability to
carry out systematic model parameter space explorations is critical for examining model sensitivities and uncertainties under different plausible scenarios. The implementation of efficient, parallel algorithms to model fluvial transport in SPGM makes model parameter space exploration more feasible and has the potential to provide new insights into the influence of dynamic topography on landscape evolution over geological time.

6. Conclusions

In this paper, we present the implementation of a parallel, multi-process [27] numerical model, where physical processes that contribute to mass redistribution are integrated separately to record geomorphological evolution. The modular structure of the code, with clearly defined interfaces between mesh generation, implementation of optimized numerical algorithms and output generation is particularly geared toward ease of adaptability for studying a wide range of geomorphological scenarios. We present simple examples of geomorphological evolution that demonstrate the capabilities of the code while providing guidelines for setting up more complex models for exploring the influences of forcing functions that can vary in space–time.

Acknowledgments

MG was supported by Statoil ASA, NSF EAR - 1358646 and EAR - 1247022. R.D.M was supported by Statoil ASA through ARC IH130200012.

Appendix A

Table A.1 lists physical parameters associated with each module, along with their corresponding names in parameter files. Additionally, annotated parameter-files and initial topographies for models used to produce Figs. 3–5 can be found online in Supplementary Data.

Appendix B

B.1. Verification of nonlinear diffusion algorithm

The Method of Manufactured Solutions (MMS) [28,29] offers a convenient way to verify the accuracy of implementations of numerical algorithms. A major advantage afforded by MMS is the ability to verify codes with nonlinearities for which analytical solutions are not readily available. In the generally applicable version of the method, one simply includes in the code a source term, S, that originates from the choice of a non-trivial, but analytical solution, which also defines boundary conditions—see Roache [29] for a detailed review of the procedure. In other words, one starts with an analytical solution and derives S by passing the solution through the governing equation, in order to satisfy the solution.

\[
    z = \frac{e^{-x^2-y^2}}{\pi} \tag{B.1}
\]

\[
    K_d = \frac{\sin^2(x) + \cos^2(y) + 1}{e^{\frac{\pi}{2}}} \tag{B.2}
\]

\[
    S = -4x^2 \left( \sin^2(x) + \cos^2(y) + 1 \right) e^{-x^2-y^2}
    + 4x e^{-x^2-y^2} \sin(x) \cos(x) e^{\frac{\pi}{2}}
    - 4y^2 \left( \sin^2(x) + \cos^2(y) + 1 \right) e^{-x^2-y^2}
    - 4ye^{-x^2-y^2} \sin(y) \cos(y) e^{\frac{\pi}{2}}
    + 4 \left( \sin^2(x) + \cos^2(y) + 1 \right) e^{-x^2-y^2} - \frac{e^{-x^2-y^2}}{\lambda e^{\frac{\pi}{2}}} \tag{B.3}
\]

For our purposes, we choose Eq. (B.1) as the analytical solution to Eq. (8) and apply Eq. (B.2) as the space–time varying diffusivity field to derive S (Eq. (B.3)), using the symbolic mathematics package, SymPy [30]. We add S to the right hand side of Eq. (8) as the source term and solve the resulting equation numerically for comparison against the analytical solution. Fig. B.7 shows the evolution of the numerical solution, with \( \lambda = 10 \), which is an arbitrary constant, in Eqs. (B.1)–(B.3). Fig. B.8A shows that the numerical solution agrees well with the analytical one and Fig. B.8B shows the rate of convergence as a function of grid spacing.

Appendix C

C.1. Effects of plume-induced dynamic topography

We combine the model in Section 3.5 with a transient uplift function that represents the motion of a mantle plume beneath
Fig. B.7. Evolution of the numerical solution described in Appendix B.

Fig. B.8. (A) Comparison of the numerical and analytical solutions as described in Appendix B. (B) Squared norm of the difference between numerical and analytical solutions, as a function of grid spacing \( \Delta x \), is shown for the same problem as described in Appendix B, but for a range of grid resolutions.

Fig. C.9. Final topography (top-left), cumulative erosion and deposition patterns (top-right), river networks in map-view (bottom-left) and cumulative uplift (bottom-right) resulting from an upwelling plume (see text in Appendix C.1) are shown at 20 × 10^6 years. A factor of 10 scaling is applied to the 3D perspective views to exaggerate topography.

continental lithosphere (e.g. Braun et al. [31]. Laboratory models of mantle plumes suggest that initiation of plumes involve a phase of rapid uplift, followed by a more gradual decline [32]. Here we approximate the domal rate of uplift (representing dynamic topography from an upwelling mantle plume) by a Gaussian function, centered at the plume axis. The center of the plume moves along a near-diagonal path across the domain and the associated rate of uplift initially increases, followed by a gradual waning of both strength and radial extent of influence. Fig. C.9 shows various model attributes after a 20 Myr period of evolution and animation S1 demonstrates the significant influence that plume-related dynamic topography can exert on the evolution and reorganization of drainage networks that in turn influence patterns of erosion and sediment deposition through time. Preexisting topography evolves at a slower rate compared to regions influenced by the plume, suggesting that our results are qualitatively in agreement with those presented in Braun et al. [31], where it was shown that erosion of dynamic topography increases linearly with its wavelength. It is also important to note that drainage reorganization occurs predominantly in downstream channel networks where channel gradients are gentler, whereas upstream channel networks remain mostly intact over the last 10 Myr period (animation S1). This is likely to have implications for drainage reversals in river systems, induced by transient plume-related dynamic topography.
C.2. Transition from detachment-limited to transport-limited scenario

In order to demonstrate the contrasting effects of detachment-limited and transport-limited modes of fluvial erosion, we take the initial topography and dynamic uplift history from the model described in Appendix C.1 and parameterize two stages of drainage network evolution. During the first 10 Myr period we impose a detachment-limited state, where all eroded material is removed from the computational domain. Drainage networks then switch to a transport-limited state, potentially due to a change in climatic conditions and eroded sediments are deposited downstream over the subsequent 10 Myr period. This is achieved by running the model using two separate configuration files, each spanning a 10 Myr period of model evolution—the final topography from the first stage is used as an input to the next. Fig. C.10 shows instances of topography at 5 Myr intervals and animation S2 demonstrates the sharp contrast in the morphology of drainage networks over the two separate stages of evolution. During the first 10 Myr period, the plateau front is rapidly eroded away, whereas over the next 10 Myr period, incisive drainage networks reach deep into the interior of the plateau.

While river systems are likely to transition between states more gradually, this example serves to demonstrate that such transitions can be modeled in SPGM and that they can be made more gradual by cascading several intermediate stages, which would feature a varying $K_f$ to effect a smoother transition.

Appendix D. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2018.07.005.

References

[1] Braun J. The many surface expressions of mantle dynamics. Nat Geosci 2010;3(12):825–33.
[2] Beaumont C, Fullsack P, Hamilton J. Erosional control of active compressional orogens. In: Thrust tectonics. Springer; 1992, p. 1–18.
[3] Braun J, Sambridge M. Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization. Basin Res 1997;9(1):27–52.
[4] Willett SD. Orogeny and orography: the effects of erosion on the structure of mountain belts. J Geophys Res: Solid Earth (1978-2012) 1999;104(B12): 28957–81.
[5] Garcia-Castellanos D. Interplay between lithospheric flexure and river transport in foreland basins. Basin Res 2002;14(2):89–104.
[6] Simpson G, Schlunegger F. Topographic evolution and morphology of surfaces evolving in response to coupled fluvial and hillslope sediment transport. J Geophys Res: Solid Earth (1978-2012) 2003;108(B6).
[7] Simpson G. A dynamic model to investigate coupling between erosion, deposition, and three-dimensional (thin-plate) deformation. J Geophys Res: Earth Surf (2003–2012) 2004;109(F2).
[8] Simoes M, Braun J, Bonnet S. Continental-scale erosion and transport laws: a new approach to quantitatively investigate macroscale landscapes and associated sediment fluxes over the geological past. Geochim Geosyst 2010;11(9).
[9] Braun J, Willett SD. A very efficient O(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution. Geomorphology 2013;180:170–9.
[10] Wallis C, Watson D, Tarboton D, Wallace R. Parallel flow-direction and contributing area calculation for hydrology analysis in digital elevation models. Power 2009;11(8):7.
[11] Do HT, Limet S, Melin E. Parallel computing flow accumulation in large digital elevation models. Procedia Comput Sci 2011;4:2277–86.
[12] McGough A, Liang S, Rapoportas M, Grey R, Vinod GK, Maddy D, Trueman A, Wainwright J. Massively parallel landscape-evolution modelling using general purpose graphical processing units. In: High Performance Computing (HiPC), 2012 19th International Conference on. IEEE; 2012. p. 1–10.
[13] Qin CZ, Zhan L. Parallelizing flow-accumulation calculations on graphics processing units — From iterative DEM preprocessing algorithm to recursive multiple-flow-direction algorithm. Comput Geosci 2012;43:7–16.
[14] Stroustrup B. The C++ Programming Language. fourth ed. Addison-Wesley Professional; 2013.
[15] Guibas L, Stolfi J. Primitives for the manipulation of general subdivisions and the computation of voronoi. ACM Trans Graph 1985;4(2):74–123.
[16] Lischinski D. Incremental delaunay triangulation. Graphics gems IV 1994;47–59.
[17] Freeman TG. Calculating catchment area with divergent flow based on a regular grid. Comput Geosci 1991;17(3):413–22.
[18] Squalacote AH, Ahrens J. The paraview guide vol. 366. Kitware; 2007.
[19] Whipple KX, Tucker GE. Implications of sediment-flux-dependent river incision models for landscape evolution. J Geophys Res: Solid Earth (1978-2012) 2002;107(B2):ETG–3.
[20] Gasparini NM, Whipple KX, Bras RL. Predictions of steady state and transient landscape morphology using sediment-flux-dependent river incision models. J Geophys Res: Earth Surf (2003–2012) 2007;112(F3).
[21] Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical recipes: the art of scientific computing, vol. 2. Cambridge University Press London; 1987.

[22] Kooi H, Beaumont C. Escarpment evolution on high-elevation rifted margins: insights derived from a surface processes model that combines diffusion, advection, and reaction. J Geophys Res: Solid Earth (1978-2012) 1994;99(B6):12191–209.

[23] Tucker GE, Slingerland RL. Erosional dynamics, flexural isostasy, and long-lived escarpments: a numerical modeling study. J Geophys Res: Solid Earth 1994;99(B6):12229–43.

[24] Perlin K. An image synthesizer. ACM Siggraph Comput Graph 1985;19(3):287–96.

[25] Zienkiewicz OC, Taylor RL, Zienkiewicz OC, Taylor RL. The finite element method vol. 3. McGraw-hill London; 1977.

[26] Guennebaud G, Jacob B. et al., Eigen v3 http://eigen.tuxfamily.org.

[27] Pazzaglia FJ. Landscape evolution models, The Quaternary Period in the United States. Developments in Quaternary Science vol. 1. pp. 247–274.

[28] Knupp P, Salari K. Verification of computer codes in computational science and engineering. CRC Press; 2002.

[29] Roache PJ. Code verification by the method of manufactured solutions. J Fluids Eng 2002;124(1):4–10.

[30] SymPy Development Team. SymPy: Python library for symbolic mathematics, 2016, http://www.sympy.org.

[31] Braun J, Robert X, Simon-Labric T. Eroding dynamic topography. Geophys Res Lett 2013;40(8):1494–9.

[32] Griffiths R, Gurnis M, Eitelberg G. Holographic measurements of surface topography in laboratory models of mantle hotspots. Geophys J Int 1989;96(3):477–95.