

A simple algorithm for the mapping of TIN data onto a static grid: Applied to the stratigraphic simulation of river meander deposits

Quintijn Clevis^{a,*}, Gregory E. Tucker^a, Stephen T. Lancaster^b, Arnaud Desitter^c,
Nicole Gasparini^d, Gary Lock^e

^aDepartment of Geological Sciences, 2200 Colorado Av., Boulder, USA

^bDepartment of Geosciences, Oregon State University, Corvallis, USA

^cSchool of Geography and the Environment, Oxford University, Oxford, UK

^dDepartment of Geology and Geophysics, Yale University, New Haven, USA

^eDepartment of Archaeology, Oxford University, Oxford, UK

Received 22 February 2005; received in revised form 29 April 2005; accepted 24 May 2005

Abstract

Triangulated irregular networks (TIN) in landscape evolution models have the advantage of representing geologic processes that involve a horizontal component, such as faulting and river meandering, due to their adaptive remeshing capability of moving, adding and deleting nodes. However, the moving node feature is difficult to integrate with the accumulation of a three-dimensional (3D) subsurface stratigraphy, because it requires 3D subsurface interpolation, which results in stratigraphic data loss due to heterogeneity of the subsurface and averaging effects. We present a simple algorithm that maps any changes in the configuration of TIN landscape nodes onto a static grid, facilitating the creation of a fixed stratigraphic record of TIN surface change. The algorithm provides a practical solution not only for the stratigraphic problem, but also for other problems that involve linking of models that use TIN and raster discretization schemes. An example application is presented using the river meandering module incorporated in the CHILD landscape evolution model. Examples are shown of cross-sections, and voxel distributions and geo-archaeological depth–age maps. These illustrate the type of insights that can be obtained from process-based modeling of subsurface fluvial architecture, and highlight potential applications of stratigraphic simulation.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Adaptive TIN; Meander; Stratigraphic simulation; Geoarchaeology; Landscape evolution; Geomorphology; Streams

1. Introduction

Surface process models are widely used in geomorphology and geology, and the developments

in the field follow each other rapidly. Much of the progress consists of the improvement of existing models, such as the addition of new surface processes (e.g., Densmore et al., 1998), new sediment transport algorithms (e.g., Gasparini et al., 1999), the recording of stratigraphy (e.g., Johnson and Beaumont, 1995; Clevis et al., 2004a) or

*Corresponding author. Tel./fax: +313 025 35119.

E-mail address: quintijn.clevis@shell.com (Q. Clevis).

applying realistic scenarios with stochastic rainfall (e.g., Tucker and Bras, 2000; Karszenberg, 2002). Advances have also been made in improving the backbone of surface-process models by changing the spatial discretization of the model landscape and the method by which water and sediment are routed over the surface. This has resulted in a new generation of models based on a self-adapting irregular triangular network (TIN) as opposed to the commonly used static rectangular grid.

Two notable examples applicable on the geologic timescale are CASCADE (Braun and Sambridge, 1997) and CHILD (Tucker et al., 2001a, b). In these models the nodes representing the landscape surface are connected to each other using Delaunay triangulation (Tipper, 1991). Delaunay triangulation is a well-known method in computational geometry and it has been widely applied in modeling solid objects and in constructing finite-element meshes. However, many of these applications use triangulation in a static way, in the sense that the object of interest is discretized only once or a limited number of times during a simulation. This is not the case in the two surface process models referred to here. They are able to constantly update their landscape representation over geologic time by adding, subtracting, and/or moving nodes and updating the triangulation accordingly.

The main advantage of an irregular adaptive mesh for geological applications is the capability to describe changing surface patterns that demand a high degree of geometrical flexibility, such as horizontal tectonic transport associated with thrusting (Miller and Slingerland, 2003), normal-faulting, strike-slip motion (Braun and Sambridge, 1997), and river meandering (Tucker et al., 2001b). For example, the CHILD model contains a module for river meandering that exploits the adaptive remeshing capabilities to simulate gradual channel migration (Lancaster and Bras, 2002; Lancaster, 1998). An additional feature of the CHILD model is active sediment sorting and the storage of a three-dimensional (3D) subsurface stratigraphy (Tucker et al., 2001b).

The use of a TIN-based mesh with movable nodes to simulate channel migration, however, introduces a subsurface storage and information retrieval problem. This arises because the surface nodes constantly change their position relative to the (fixed) stratigraphic information, which is locked in the underlying sedimentary substrate. In other words, the nodes are intended to represent mor-

phologic features (in this example, the centerline of a river channel) that can move horizontally relative to the underlying rock/sediment mass. The mobility of these landform-based elements complicates the process of storing and retrieving information about the underlying stratigraphy. In order to facilitate stratigraphic information retrieval, there are three potential strategies that might be used. Each of these is unavoidably affected by loss of stratigraphic information due to interpolation procedures, and each involves different trade-offs between efficiency and information fidelity:

Method 1. All stratigraphic information is directly coupled to the (potentially) mobile nodes in the surface TIN.

Method 2. Stratigraphic information is recorded in a fixed subsurface TIN that underlies the mobile surface TIN.

Method 3. Stratigraphic information is recorded in a fixed subsurface rectangular grid that underlies the mobile surface TIN.

When stratigraphic information is directly linked to moving nodes in the TIN (method 1), frequent 3D stratigraphic interpolation is needed in order to retrieve the subsurface information at the newly assigned node locations. A pseudo-code example of a suitable algorithm that was previously applied in the CHILD model is given in Appendix A. The mechanism of the algorithm is similar to the process of shuffling a deck of cards. Lithological information (the cards) is stored in a series of layers stacked beneath each node. Each layer represents a particular time interval, and contains lithologic attributes such as grain size composition. When a node moves or a new node is added, the layer information is interpolated based on the surrounding nodes, and the layers are restacked in stratigraphic–chronological order. When surface nodes move distances comparable to the size of the model domain, the repeated re-shuffling can lead to significant information loss. Experience has shown that this ‘layers-follow-nodes’ method can lead to three problems when applied to a TIN mesh containing a large number of mobile nodes. First, subsurface information is partially lost due to the combined effect of a strong lateral stratigraphic heterogeneity, averaging effects, and progressive propagation of these effects through the subsurface as stratigraphic columns are recreated from columns interpolated in previous time steps. A simplified 2D example of the error

propagation is given in Fig. 1, where the correct original stratigraphic heterogeneity and age information are lost and ‘smeared’ out in the direction of channel movement. A second disadvantage of the method is that it produces numerous thin layers (representing short time intervals of deposition), which are advected into neighboring columns, increasing the demand on computer memory but without contributing significantly to stratigraphic volume or fidelity. Finally, the method involves frequent 3D interpolation of large datasets (~up 250 layers per node), which harms performance as interpolation is done during the already computationally intensive routine of channel migration and TIN remeshing.

The problem of repeated shuffling and information smearing can be resolved by adding a second, fixed mesh to store the subsurface information, and incrementally mapping surface erosion/deposition events onto the fixed subsurface mesh (methods 2 and 3). Obviously, the incorporation of a second stratigraphic storage mesh increases the demands on computer memory, but for the geological applications we describe here, computation time rather than storage is the limiting constraint (as is true for many types of dynamic model). This approach still involves spatial averaging because of the need to map information from one tessellation onto another, but unlike method 1, the wavelength of the resulting information diffusion is limited to that of the mesh resolution. In order to minimize information loss at this local scale, the resolution of the static mesh should be at least equal to or higher than that of the surface mesh.

One could accomplish this dual-mesh method using either a pair of TIN meshes (method 2) or a mobile surface TIN coupled to a subsurface raster grid (method 3). Using a TIN-based data structure for stratigraphic storage introduces the need for a relatively complicated routine for interpolating and mapping surface changes, because geometric elements in both meshes mismatch due to the mobile character of the nodes in the surface mesh. This could be avoided by using an adaptive subsurface TIN data structure that dynamically subdivides into smaller polygons while tracking the changing geometry of the TIN at the surface. However, this approach would involve permanently adding nodes to the subsurface TIN and would require a time-intensive 3D interpolation procedure similar to the one presented in Appendix A in order to retrieve the stratigraphy at different subsurface mesh locations.

Use of a regular (raster) fixed grid (method 3) avoids extra computational-geometry overhead and allows for efficient high-resolution stratigraphic storage. We present a straightforward and fast algorithm for communicating and interpolating between a dynamically evolving TIN landscape surface and the regular subsurface grid. The grid is less prone to data loss than an auxiliary TIN mesh, thanks to its simplicity, because it brings the benefit of easy data access and storage economy, thus allowing higher resolution and minimizing data loss. The algorithm presented translates the changes in elevation due to the shifting meander channel and general erosion and deposition from the TIN landscape incrementally into stratigraphy.

Method 3 is applied here to generate a 3D fluvial stratigraphy associated with the development of a meandering channel on Holocene timescales. The meander stratigraphy model is driven by a geologic scenario and could therefore be used by Quaternary geologists, groundwater modelers and archaeologists to better understand fluvial architecture and the subsurface distribution of archaeological-important sedimentary units (e.g., Clevis et al., *in press*). In addition, the mapping algorithm we describe can also be used in solving a range of problems that involve information transfer between TIN and raster formats. There is an increasing trend toward coupling different types of model in order to deepen understanding of coupled Earth systems. Examples include coupled surface and groundwater models, coupled deep-water and shoreface hydrodynamic models, and coupled surface-process and thermo-mechanical deformation models (e.g., Beaumont et al., 1992). Often, there are cases in which mapping of two different mesh types is unavoidable, as in the case of connecting a TIN-based finite element model of groundwater dynamics with a raster-based surface-water model. Indeed, developing technology for coupling models with varying space- and time-discretization schemes is one component of the recent effort to develop a Community Surface Dynamics Modeling System (Slingerland et al., 2002). It is one of the problems anticipated for solving “engineering time-scale” sediment transport problems on adaptive meshes with spatially varying properties of the sediment substratum. For problems that involve a necessary marriage of TIN-based and raster-based models, our approach offers a practical and simple solution.

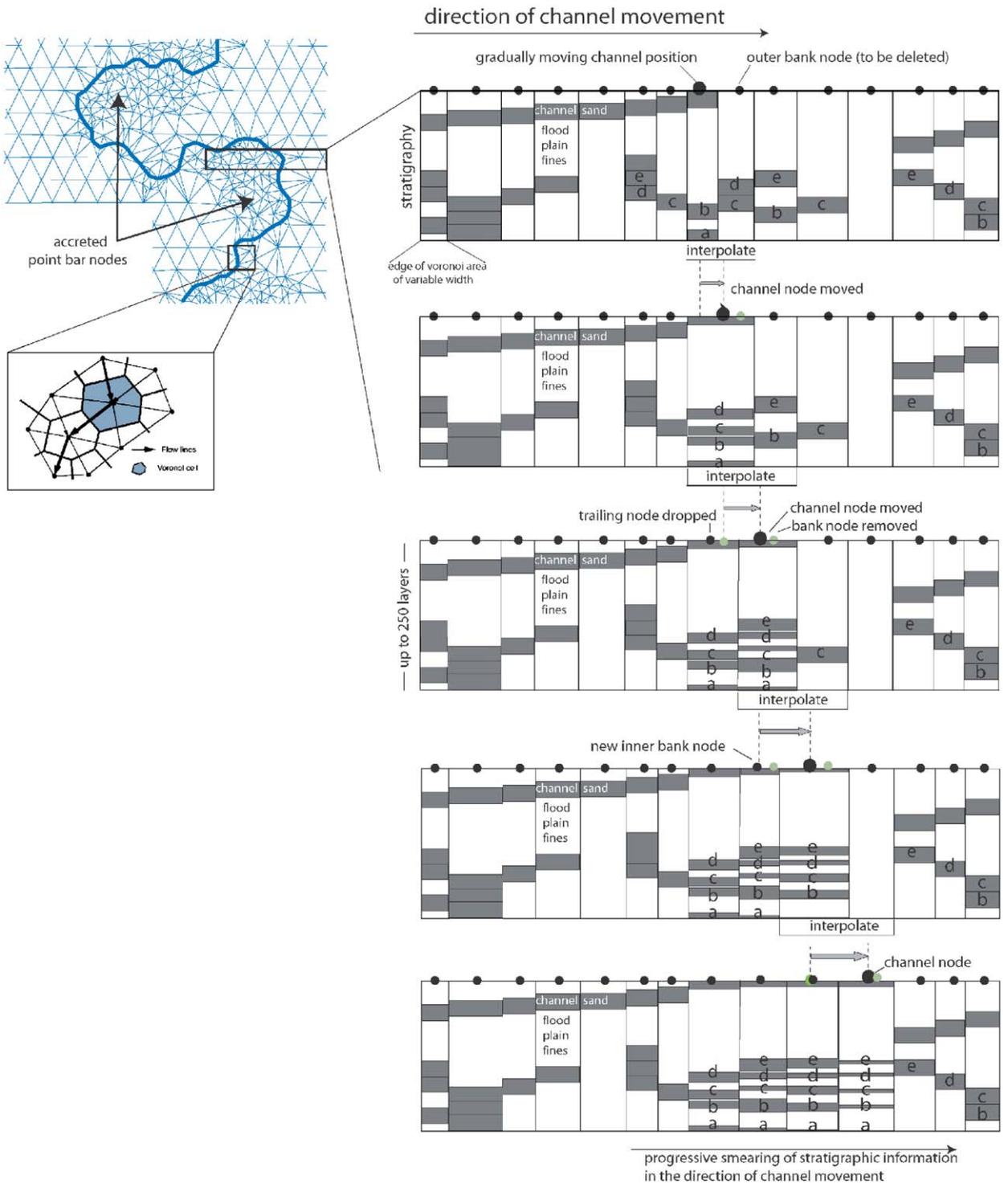


Fig. 1. Planform example of river meandering simulated on a Delaunay triangulated mesh in CHILD, illustrating problems that arise when stratigraphic information is coupled to moving surface nodes. Mesh densification represents accretion of point bar nodes that trail moving position of main channel. Coupling stratigraphy directly to these movable TIN nodes generates need to retrieve stratigraphic information for moving nodes by interpolation from surrounding nodes. Layers with identical ages represented by letter tags are re-interpolated each time a node moves. As a result, information diffuses over long distances through subsurface, blurring original stratigraphic heterogeneity and adding many new thin layers. Actual interpolation process in 3D is much more complicated, computationally intensive, and prone to introduction of stratigraphic interpolation errors as channel position sweeps back and forth over landscape surface (see Appendix A).

2. Meandering module in the CHILD model

CHILD is a model of landscape erosion, sediment sorting and topographic evolution based on an adaptive triangular mesh (Tucker et al., 2001a, b) and contains modules for stream meandering and stratigraphy (Fig. 2). In contrast to the previous model descriptions of meandering that use depth-averaged continuity and momentum equations (Johannesson and Parker, 1989; Sun et al., 1996), the meander module in CHILD uses a combination of simplified process-physics and rules, in order to operate at geologic time-scales (Lancaster and Bras, 2002; Lancaster, 1998). It reproduces observed features of meandering such as the formation of

compound bends. The meander model is based on the concept of ‘topographic steering’ (Dietrich and Smith, 1983; Smith and McLean, 1984), which derives from the observation that secondary flows over the bed topography translate the erosive high-velocity core in a channel segment laterally. This secondary flow results in transfer of momentum and maximum shear stress towards the outer bank and downstream, causing erosion. The rate at which meander channel nodes in CHILD are allowed to migrate is defined by the rate of bank erosion, which is proportional to this bank shear stress.

$$R_{migration} = E\tau\hat{n}, \tag{1}$$

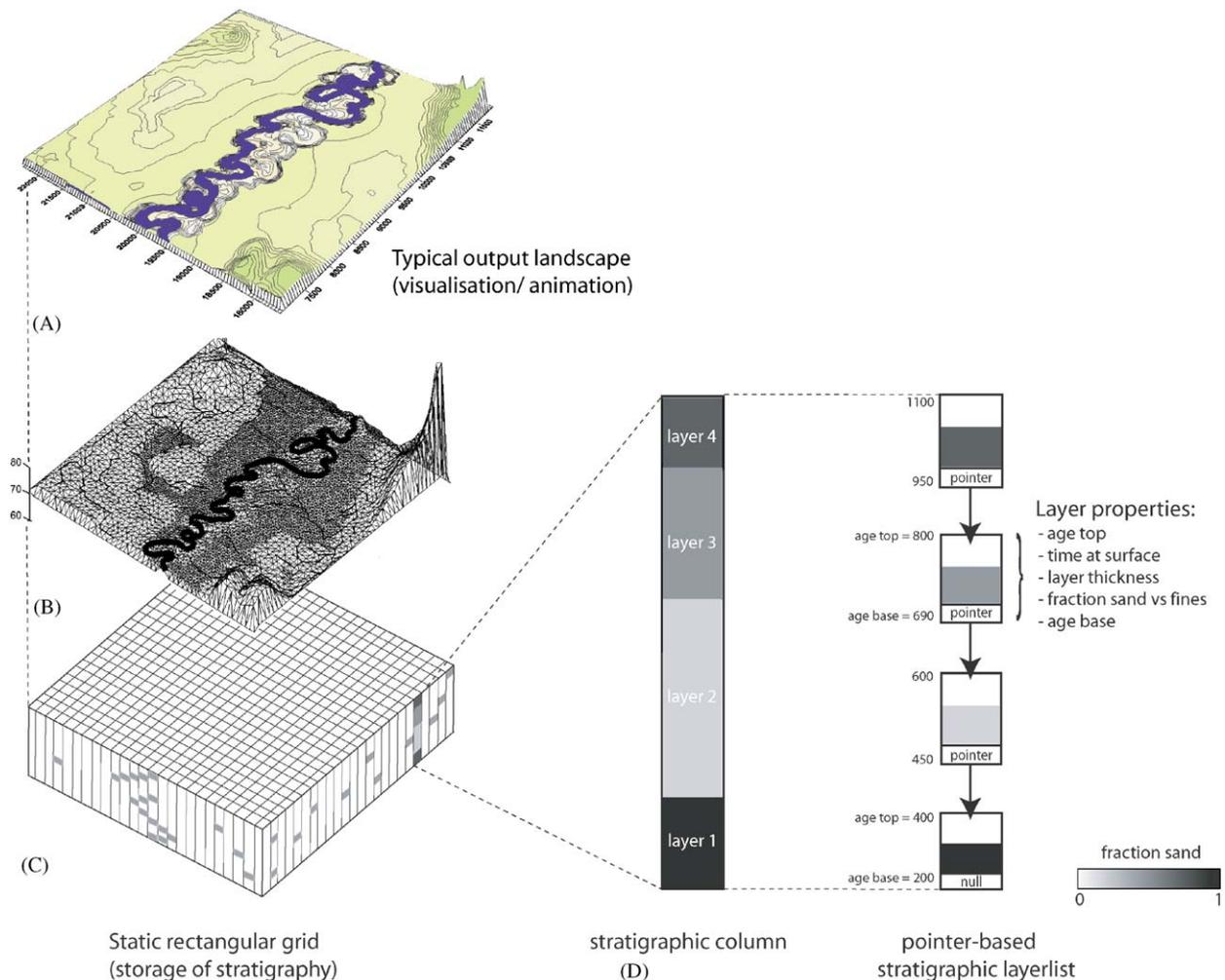


Fig. 2. (A) Typical output landscape of CHILD meander module. (B) Computational mesh used to dynamically move, add and delete points in response to lateral movement along main channel. (C) Static rectangular grid used for storage of subsurface stratigraphy. (D) Stratigraphic columns are stored as linked lists of layer packages coupled to static grid positions (Oualine, 1997). Layer packages contain information about layer age and grain size texture.

where E is the bank erodability coefficient, τ is the bank shear stress and \hat{n} is the unit vector perpendicular to the downstream direction. The adaptive remeshing capability incorporated in CHILD is especially designed to simulate the gradual development of complex-shaped meander bends over time. During the meandering process, the central channel nodes are iteratively relocated, any obstructing outer bank nodes are deleted, and additional nodes are added behind the former channel position in order to create point bars. As a result of channel movement the floodplain mesh is continuously updated. Meander segments that approach one another are able to link in order to create neck cut-offs if a slope advantage exists. The model does not simulate the effect of chute cut-offs or large-scale avulsions.

Besides active channel meandering, CHILD simulates another process characteristic for a fluvial system: floodplain deposition. The method applied for floodplain deposition is based on the observation that floodplain sedimentation rate decreases exponentially with distance from the main channel (Pizzuto, 1987; Howard, 1992, 1996; Mackey and Bridge, 1995). Local deposition depth in a single flood event is calculated as

$$dH_{\text{overbank}} = \mu(W - z)e^{(-d/\lambda_{ob})} \Delta t, \quad (2)$$

where dH_{overbank} is the depth of sediment deposited during a flood event at a floodplain node, μ is the deposition rate constant [L/T], λ_{ob} is the distance-decay constant at which overbank sedimentation rate decreases to zero, and d is the distance between the active floodplain node and the nearest meander channel node. The floodplain model takes into account the local topography by incorporating the difference between floodwater height W and the local floodplain elevation z . The maximum flood height W and the duration flood events, Δt , are driven by a series of storms, which are drawn at random from an exponential frequency distribution with user-specified mean values (Tucker and Bras, 2000).

3. TIN to grid mapping algorithm

A CHILD simulation begins with the creation of a starting TIN landscape mesh, and the initialization of several objects responsible for input, output, drainage network organization and choice of sediment transport algorithms and variables (Fig. 3A). The modules for meandering and stratigraphy

```

int 'CHILD' main()
{
  //(A) Open input file and initialise objects and class-types;

  tInputFile inputFile();
  tMesh<TIN_Node> TINmesh();
  tLOutput<TIN_Node> output();
  tStorm storm();
  tStreamNet strmNet();
  tErosion erosion();

  //(B) Create meander-floodplain objects, if applicable

  if( optFloodplainDep ) new floodplain(inputFile, &TINmesh);
  if( optMeander ) new strmMeander(inputFile, &TINmesh);
  if( optStratGrid ) new tStratGrid(inputFile, &TINmesh)
  output.WriteOutput( time = 0.);
  cout << "Initialization done.\n";

  //(C)----- MAIN TIME LOOP-----
  while( ! time.IsFinished() )
  {

    time.ReportTimeStatus();

    //Get a new storm duration and intensity
    storm.GenerateRandomStorm();

    // Re-calculate drainage paths
    strmNet.UpdateStreamNet();

    // Do sediment transport and change TIN_Node elevations
    erosion.ErosionAndDeposition();
    if( optStratGrid ) stratGrid->UpdateStratGrid(0, time.getT());

    // Compute channel displacement and adapt TINmesh
    if( optMeander ) strmMeander->MigrateChannel( time.getT());
    if( optStratGrid ) stratGrid->UpdateStratGrid(1, time.getT());

    if(optFloodplainDep) floodplain->DepositOverbank();
    if( optStratGrid ) stratGrid->UpdateStratGrid(0, time.getT());

    // Advance time by incrementing storm durations
    time.Advance( storm.getStormDuration() + storm.interstormDuration());

    if( time.CheckOutputTime() ) output.WriteOutputFiles( time.getT());

  } // -----end of main time loop-----

  //(D) delete created objects
  delete objects();
}

```

Fig. 3. Code summary of CHILD main routine, giving an outline of model procedure. First classes and objects dealing with adaptive mesh, drainage network, sediment transport, and floodplain are created. 'UpdateStratGrid' functions which are responsible for updating stratigraphy are called after functions for channel migration and erosion and deposition in MAIN time loop.

discussed here are optional, and they are activated using 'flags' in the main input file (Fig. 3B). Simulations are performed using a time loop in which each surface-process function is called sequentially in incremental small time steps (Fig. 3C). The length of the time steps and the intensity of the associated rainfall are dictated by a storm-generation module, which generates a series of storms at

random from a Poisson probability distribution (Eagleson, 1978).

The meander process results in changes in the configuration and elevation of the TIN nodes. These changes are transferred or mapped to the static stratigraphy grid, called ‘StratGrid,’ from several locations within the CHILD main loop (Fig. 3). The responsible transfer function ‘UpdateStratGrid’ is called after the main surface process functions ‘ErosionAndDeposition’, ‘MigrateChannel’ and ‘DepositOverbank’. The function ‘UpdateStratGrid’ calls ‘UpdateConnect’ and ‘ChangeStrat-

Grid’. UpdateConnect contains a loop over the triangles of the TIN mesh, and it links static grid node locations to the closest triangle (Fig. 4A). For every triangle an enveloping area ‘thisBox’ is determined by finding the maximum and minimum x and y coordinates of the corners $P(0)$, $P(1)$ and $P(2)$ of the triangle (Fig. 4B). These bounding x , y coordinates are then translated into a range of StratGrid indices $imin$, $imax$ and $jmin$, $jmax$. In cases where the actual dimensions of the StratGrid differ from the dimensions of the TIN mesh, these indices are clipped to indices $bimin$, $bimax$ and

```

void tStratGrid::updateConnect()
{
    // Nullify/ Reset the stratGrid connection matrix
    // 'StratConnect' which contains a pointer to a TIN
    // triangle for every i,j location of the stratGrid
    {
        for(int i=0; i<imax; ++i)
            for(int j=0; j<jmax; ++j)
                (*StratConnect)(i,j) = NULL;
    }

    // Loop over the TIN Triangles by iterating over a linked list
    // 'trilter' containing pointers to the Triangles in the mesh
    TIN_Triangle *ct;
    TINMesh< TIN_Node >::TriangleListIterator trilter( mp->getTriList() );
    for( ct = trilter.FirstP(); !trilter.AtEnd(); ct = trilter.NextP() ) {

        // Find dimensions initial box containing the current triangle
        const TriBox thisBox( ct,xcorner,ycorner,griddx );
        // triangle does not contain any stratNode
        if (thisBox.containsNone())
            continue;

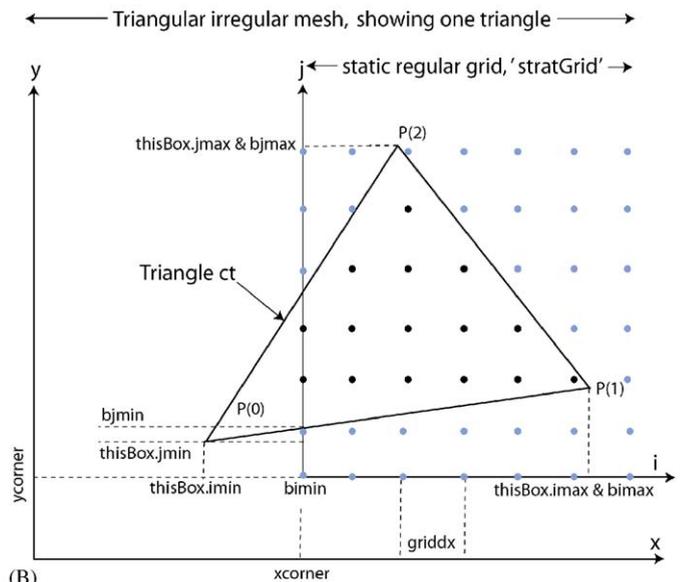
        // Set StratConnect for the stratNode within the current triangle
        {
            // clip bounds within the actual bounds of the StratGrid
            const int bimin = max(0,thisBox.imin);
            const int bimax = min(getImax()-1,thisBox.imax);
            const int bjmin = max(0,thisBox.jmin);
            const int bjmax = min(getJmax()-1,thisBox.jmax);

            // Find which StratNode is contained within the current triangle.
            for(int i=bimin; i<=bimax; ++i) {
                for(int j=bjmin; j<=bjmax; ++j) {
                    if ((*StratConnect)(i,j) != NULL)
                        continue;

                    // Positive, assign i,j a pointer to the Triangle
                    if (ct->containsPoint( (*StratNodeMatrix)(i,j).getX(),
                                           (*StratNodeMatrix)(i,j).getY() )) {
                        (*StratConnect)(i,j) = ct;
                    }
                }
            }
        }
    }
}

```

(A)



(B)

```

bool Triangle::containsPoint(double x,double y) const
{
    const double P[] = {x,y};
    const double P0[] = {p[0]->getX(),p[0]->getY()};
    const double P1[] = {p[1]->getX(),p[1]->getY()};
    if (predicate.orient2d(P0,P1,P) < 0)
        return false;
    const double P2[] = {p[2]->getX(),p[2]->getY()};
    if (predicate.orient2d(P1,P2,P) < 0)
        return false;
    if (predicate.orient2d(P2,P0,P) < 0)
        return false;
    return true;
}

```

(C)

Fig. 4. Code summary for function UpdateConnect (A) and illustration of geometry involved (B). UpdateConnect selects a group of static grid nodes located within box defined by coordinates of Triangle’s corners. (C) Routine ‘Containspoint’ evaluates which of selected nodes is positioned within Triangle by computing determinant of Triangle limb vectors and node coordinates. Nodes located within Triangle are assigned a pointer to this Triangle in order to link ‘StratGrid’ to TIN mesh.

b_{jmin} and b_{jmax} , to make sure they fall within the boundaries of the StratGrid. Next, all the StratGrid locations in the box selection are checked for their position relative to the triangle, whether they are inside or outside the triangle. This is done using the function ‘ContainsPoint’ (Fig. 4C) which calls a standard geometric orientation predicate routine for every limb of the triangle (for source see: <http://www-2.cs.cmu.edu/~quake/robust.html>). This routine evaluates whether a StratGrid node lies to the left or the right for every limb of the triangle by calculating the sign of the determinant composed of the limb vector and the node coordinates. A node which yields a positive determinant for all limb vectors (e.g. $P(0) \rightarrow P(1)$, $P(1) \rightarrow P(2)$ and $P(2) \rightarrow P(0)$), is located within the triangle. When the test in ContainsPoint is positive, the specific static node location in the StratGrid is assigned a pointer to the triangle of the TIN mesh.

The actual change in elevation and sediment properties from TIN mesh to StratGrid are transferred using the function ‘ChangeStratGrid’ (Fig. 5A). The new elevation of a StratGrid node is the result of two operations, remeshing or erosion and deposition. The first case is related to channel migration where channel TIN nodes are translated, point bar nodes are added and outer bank nodes are removed, modifying the topography flanking the channel. The corresponding static grid node is raised or lowered according to the difference between its old elevation and a new elevation, which is determined by simple linear surface interpolation between the corners of the triangles describing the adapting channel (Fig. 5B). In the second case the total elevation change is composed of incremental changes in the sediment texture at the triangle corners, ‘Dh_fine’ and ‘Dh_sand’ (Fig. 5C). At present, the StratGrid module is restricted to only two grainsize classes, but with minor code adaptation an arbitrary number of grain size classes could be incorporated. The changes in the two textures are interpolated for the stratNode location and forwarded to the function ‘EroDep’, together with the tag for the status of the simulation time. This function adds the increments of both textures to increase the elevation, and updates the new top age of the stratigraphic layer. In the example (Fig. 5D) the status of the simulation time of 100 yr is used to tag the new top ages of the updated layers in both cases of erosion and deposition. The choice for the StratGrid node spacing and the stratigraphic layer thickness in the

model is defined by the user in the input file and represents a compromise between computational efficiency and desired stratigraphic resolution. The node resolution influences the potential loss of data during mapping, and in order to avoid loss of stratigraphic detail this value should be equal or higher than the resolution of the surface Delaunay mesh. However, if the desired stratigraphic information is simply a distinction between channel sands and floodplain fines, a StratGrid resolution approaching twice the TIN spacing is sufficient.

4. Examples of experiments involving stratigraphy

4.1. Floodplain landscape evolution and subsurface stratigraphy

An example of a modeled floodplain landscape is shown in Fig. 6, where the evolution of the meander channel is given in intervals of 1500 yr (see also Table 1). During the simulation the channel bed was forced upward at a fixed rate of 1 m/kyr, a value that approximates the aggradation rate of a lowland river adjusting to rates of Quaternary and Holocene sea-level rise (Törnqvist et al., 2000). The system starts as an initially straight channel superposed upon a digital terrain model of a segment of a floodplain. Minor irregularities in the initial channel pattern at 500 yr evolve to a set of complete meander bends by around $T = 3500$ yr, at which point they are on the verge of cutting off. At $T = 5000 - 6500$ yr the outline of the main channel belt is delineated by a suite of abandoned meander loops flanking the active channel, while the surrounding floodplain is gradually buried by overbank deposition and forms a slightly convex-upward topography in cross section. Contour lines within the abandoned loops show the depositional ages of the sediment laid down by the moving channel. Each contour line represents 300 yr; their even spacing indicates relatively uniform rates of channel migration (Fig. 7A). The curve illustrating the channel development over time (Fig. 7B) confirms this regularity by showing a symmetric saw-tooth curve, where gradual loop widening and sinuosity growth is periodic and punctuated by cut-off events that occur about once every 1500 yr. The mean sinuosity of the system is about 2.1, with a gradual increase over time. A similar saw-tooth pattern has been observed in other studies using numerical models of meandering and the trend has been interpreted as a form of self-organization and

```

void tStratGrid::ChangeStratGrid(int remeshing, double time)
{
    int i,j;
    // Loop over the matrix of static stratNodes...
    for(i=0;i<imax;i++){
        for(j=0;j<jmax;j++){

            // Get for every stratNode ij a pointer to the correspondingTriangle
            // and metric coordinates sx,sy
            TIN_Triangle *ct = (*StratConnect)(i,j);
            const double sx = (*StratNodeMatrix)(i,j).getX();
            const double sy = (*StratNodeMatrix)(i,j).getY();

            // Store pointers to the Triangle's TIN node corners in array P[size=3]
            if(ct != NULL){
                TIN_Node *P[3];
                int n;
                int numnodes=0;
                for(n=0;n<=2;n++){
                    P[numnodes] = static_cast<TIN_Node *>(ct->pPtr(n));
                    numnodes++;
                }

                //-----REMESHING-----
                if(remeshing){
                    // Get the elevation of the Triangle's corners and interpolate for
                    // stratNode position sx,sy
                    const tArray<double> tri_elevs(P[0]->getZ(),P[1]->getZ(),P[2]->getZ());
                    const double newz = Interpolate(sx,sy,P[0]->get2DCoords(),
                                                    P[1]->get2DCoords(),P[2]->get2DCoords(),tri_elevs);

                    // Compare the new-z,interpolated elevation with the old existing
                    // elevation for the StratNode,what is the elevation difference ?

                    tStratNode &sn = (*StratNodeMatrix)(i,j);
                    const double oldz = sn.getZ();
                    const double dh = newz-oldz;

                    // Update the elevation of the Stratnode 'sn' and its
                    // Stratigraphy using function 'Erodep'

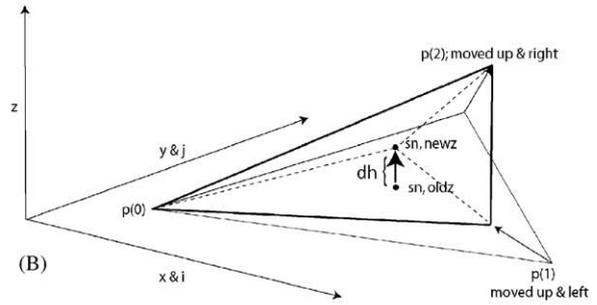
                    // -----static node goes UP-----
                    if(dh > 0.0){
                        const tArray<double> erode(dh,0.);
                        sn.EroDep(erode,time);
                    }
                    else if (dh == 0.0){
                        // do nothing...
                    }
                    // -----static node goes DOWN-----
                    else if (dh < 0.0){
                        const tArray<double> erode(0.5*dh,0.5*dh);
                        sn.EroDep(erode,time);
                    }
                }

                //-----EROSION & DEPOSITION-----
                else if(remeshing == 0){
                    // Store the incremental sand/fines changes at at Triangles
                    // corners in arrays tri_sand_dh and tri_fine_dh[size=3]
                    const tArray<double> tri_sand_dh(P[0]->getDhSand(),
                                                    P[1]->getDhSand(),P[2]->getDhSand());
                    const tArray<double> tri_fine_dh(P[0]->getDhFine(),
                                                    P[1]->getDhFine(),P[2]->getDhFine());

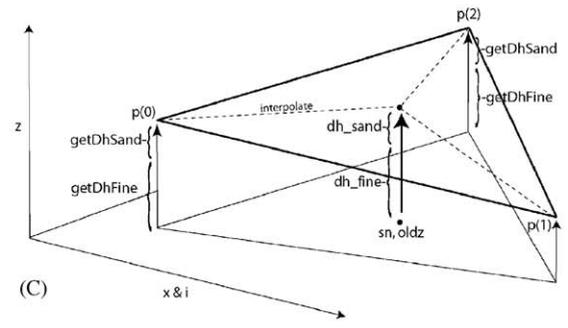
                    // Interpolate these incremental texture changes for location sx,sy
                    const double dh_sand = Interpolate(sx,sy,P[0]->get2DCoords(),
                                                       P[1]->get2DCoords(),P[2]->get2DCoords(),tri_sand_dh);
                    const double dh_fine = Interpolate(sx,sy,P[0]->get2DCoords(),
                                                       P[1]->get2DCoords(),P[2]->get2DCoords(),tri_fine_dh);

                    // Store the interpolated values in array 'erode' and call EroDep
                    const tArray<double> erode(dh_sand,dh_fine);
                    sn.EroDep(erode,time);
                }
            }
        }
    }
}
    
```

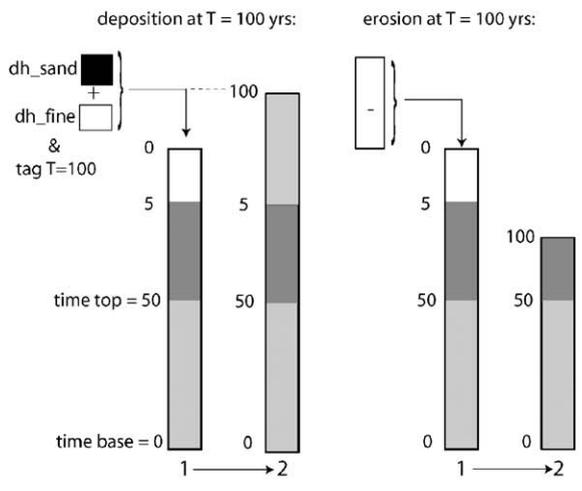
(A)



(B)



(C)



(D)

Fig. 5. Code summary for function ‘ChangeGridElevation’, where changes in elevation for stratnodes are obtained by simple linear interpolation between corners of Triangle. Nodes are updated after remeshing or after erosion and deposition (Fig. 3). In latter case incremental increases of sand and fine-grained material at Triangle corners are interpolated and applied, instead of bulk elevation change. StratNode elevations and stratigraphy are updated using function ‘EroDep’.

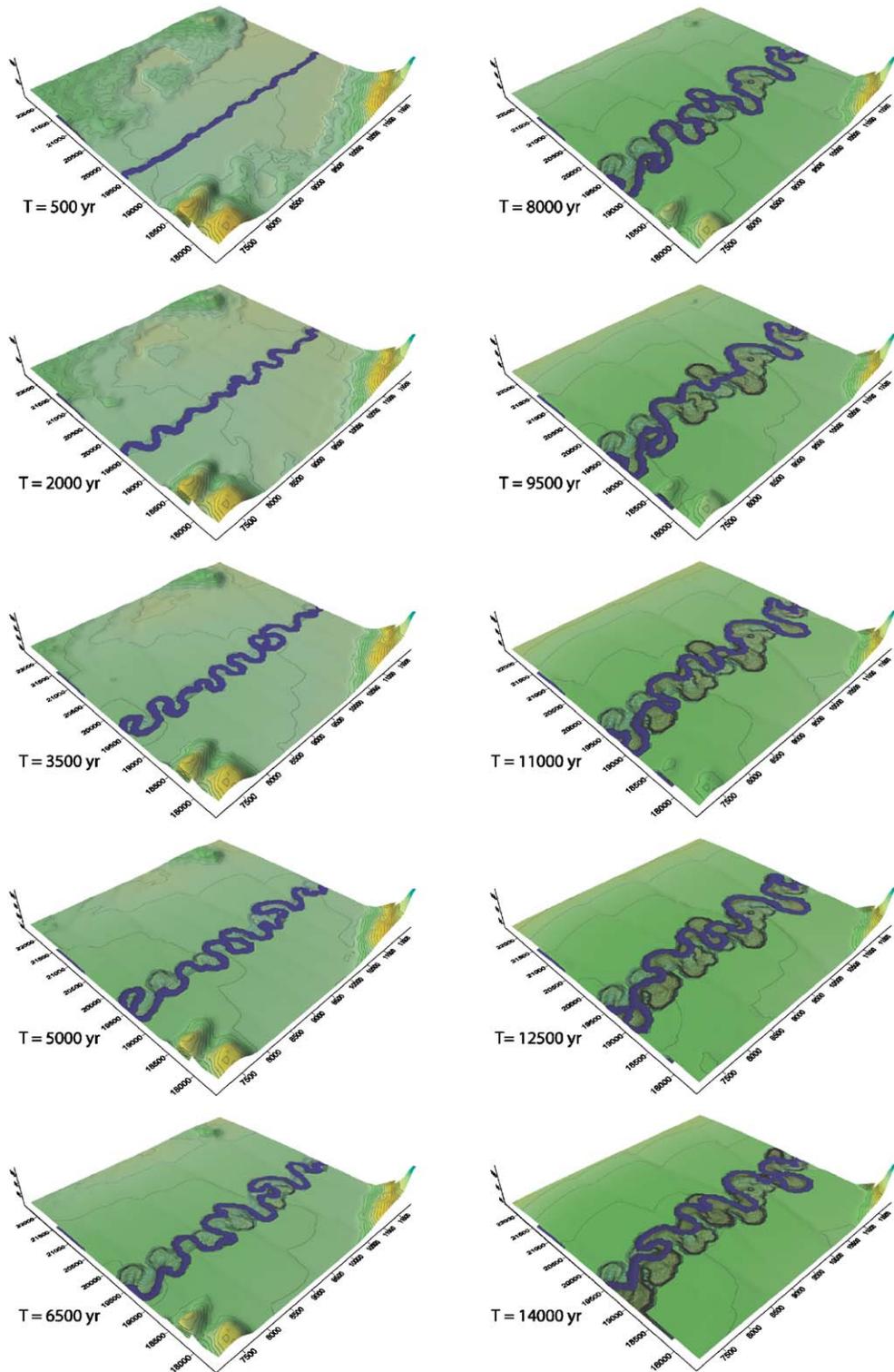


Fig. 6. Landscape snapshots illustrating development of meander pattern through time (0–14,000 yr). Contours in active channel belt indicate ages of sediment in top layer of stratigraphy. Contour interval is 300 yr. Note that relief in surrounding area is gradually buried by floodplain overbank sedimentation.

Table 1
Key variables from CHILD input file used for simulations

Symbol	Explanation	Value
E	Uniform bank erodability	0.0005 m/Nyr
μ	Overbank deposition rate constant	0.2–0.3 m/kyr
λ_{ob}	Overbank distance–decay constant	750 m
InletArea	Drainage area feeding river inlet	2000 km ²
Pmean	Storm intensity	10–12.5 m/yr
STDUR	Mean storm duration	0.06 yr
INSTDUR	Mean interstorm duration	1 yr
W_{hydr}	Resulting hydrological channel width	~50 m
Δx	TIN mesh starting cell size and StratGrid cell size	50 m

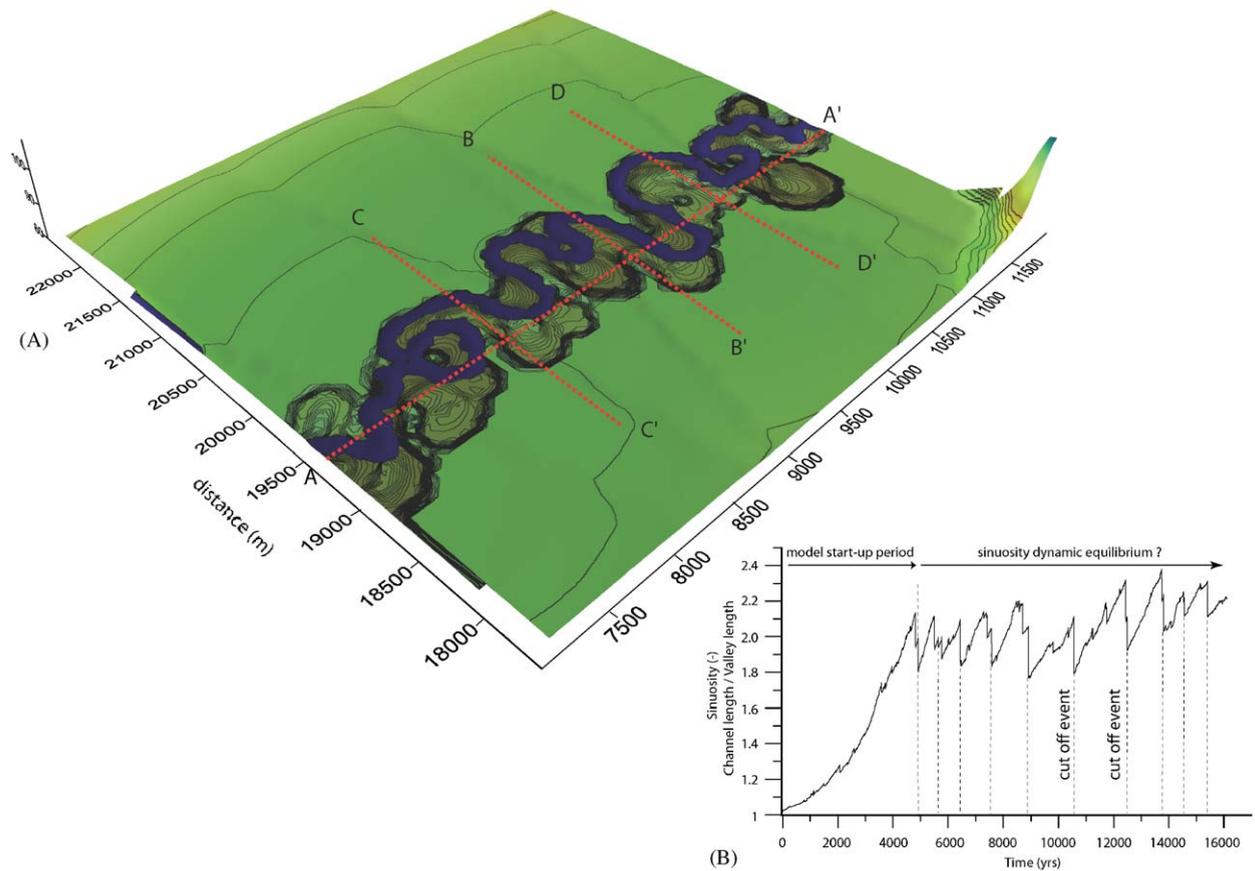


Fig. 7. Meander–floodplain system belonging to evolutionary sequence shown in Fig. 7, at $T = 15,000$ yr. Contour lines surrounding active channel reflect ages of sediment in top layer of stratigraphy. Contour interval is 300 yr. Locations of cross-sections (Fig. 9) are indicated as dotted red lines.

dynamic equilibrium in which two counteracting processes control the development of sinuosity (Stølum, 1996, 1998). Lateral migration increases the sinuosity while cut-offs decrease it. In addition, the temporal development of sinuosity is prone to a more complex feedback. Cut-offs may restore order

in chaotic, high-sinuosity channel segments, but if they occur in ordered low-sinuosity reaches they form irregularities that are consequently amplified and evolve into more high-sinuosity segments.

As a result of the frequent opening and closing of the meander loops, the sediments within the channel

belt are continuously reworked and the remnant channel pattern becomes increasingly complex through time. In order to visualize the subsurface architecture of this meander mosaic, a set of stratigraphic cross-sections was made through the simulated floodplain landscape at $T = 15,000$ yr (Fig. 8). The positions of the section lines are indicated in Fig. 7.

The cross-sections are composed of layer voxels in which the intensity of red corresponds to the fraction sand in the layers (Fig. 8). Paleo-channel positions are indicated by red voxels, whereas the floodplain fines are represented by blue voxels. The selection of the cross-section locations of and the choice of timeline spacing are part of the model's post-processing procedures that are based on scripts written in Matlab[®]. Minor incision is recognizable as dips in the timelines, reflects the channel depth. The section parallel to the direction of flow reveals

U-shaped structures, which develop when a meander loop migrates through the line of section (Fig. 8A). Initially, a loop is visible as a point at the moment it enters the line of section. As the loop grows and migrates through the section line, the upstream and downstream branch are recorded. Finally, both branches are abandoned abruptly due to an upstream cut-off and floodplain fines cover the paleo-channel positions. The U-shaped paleochannels are deflected to the right, reflecting the downstream migration of the meander bends (Figs. 8A and C). In stream perpendicular section the stratigraphic pattern is simpler (Fig. 8B). Here the section is marked by a gradually shifting paleo-channel position, followed by abrupt channel relocation in the center of the figure. Structures very similar to these, including downstream-migrating channel deposits and U-shaped bends, have been recognized in the shallow stratigraphy of

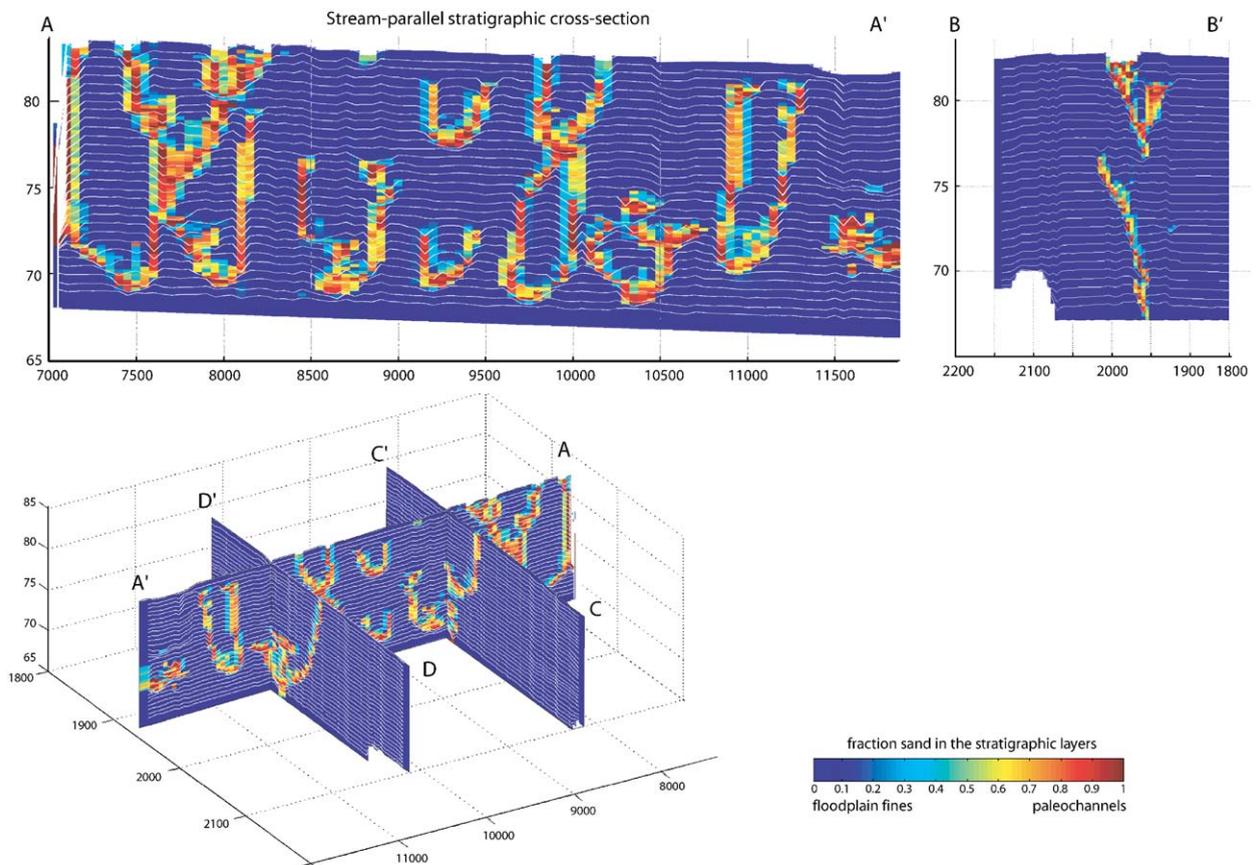


Fig. 8. Stratigraphic cross-sections through fluvial valley show subsurface distribution of paleochannels (red, sand) and floodplain fines (blue, clay). Timelines at 500-year intervals (white) in cross-sections show rapid aggradation throughout this simulation (1 m/kyr). Stream-parallel sections are dominated by U-shaped configurations of paleochannels, indicating growth of bends through line of section. Stream-perpendicular section shows lateral shifting of channel positions due to growth of loops.

submarine channel–levee systems (Posamentier, 2003; his Figs. 5, 8 and 10). This resemblance suggests a strong dynamic similarity between sub-aerial and submarine meandering channel belts. To the best of our knowledge, these subsurface features have not been predicted by previous stratigraphic forward models of river meandering.

Another way of illustrating the subsurface structure of the model floodplain is by visualizing a selection of stratigraphic layers as voxels. In Fig. 9 all layers with a sand fraction of 0.9 and larger are shown with respect to the floodplain surface and the position of the river at the end of the simulation at $T = 15,000$ yr. Clearly, these voxels are not uniformly interconnected and their distribution widens upwards towards the surface, a pattern that is indicative of the gradual widening of the channel belt due to meandering. Statistical analysis of distribution, volume and connectivity of the voxels, generated by a suite of Monte Carlo simulations, could be used to guide simulations of groundwater flow and aid predictions in aggregate exploration.

4.2. Geoarchaeology

The preservation and visibility of archaeological sites in an alluvial environment is linked to temporal and spatial patterns of channel evolution. In floodplain systems with actively migrating channels, much material can be lost to lateral bank erosion. Conversely, floodplain aggradation can bury and therefore obscure sites. Because the meander model operates at Holocene timescales, it can be used to

study the distribution of archaeological units in the subsurface, using scenario-based simulations. An example is given in Fig. 10, where the stratigraphy accumulated in two scenarios is sliced horizontally and presented as maps showing the ages of the sediment at 0.5, 1, 2 and 3 m below the floodplain. The ages of these sediments are grouped and colored according to the general archaeological timescale for the British Isles. For this example, the channel bed aggradation rate was based on the average Holocene alluvial thickness of ~ 3 m, observed in the Upper Thames Valley, UK (Needham, 1992). Two scenarios were modeled, using alternative Holocene alluviation histories typical for Western European rivers (Macklin and Lewin, 2003).

In the first scenario, both the storm intensity and the overbank sedimentation rate were increased simultaneously during distinct periods, of which the duration and relative timing correspond to the Central European cold-humid phases of Haas et al. (1998). In addition, during this simulation the channel bed level of the river was raised artificially according to a stepwise increasing curve. The trend is described by a slow channel bed aggradation rate during the Early Holocene (0.16 m/kyr) and a moderate rise since the Bronze Age (0.25 m/kyr), reflecting the onset of land use in upstream catchments (Macklin and Lewin, 2003). An even higher rate of channel bed aggradation rate (0.50 m/kyr) is applied from Roman times onwards, as this period is associated with the start of more efficient land use techniques, widespread erosion, and high

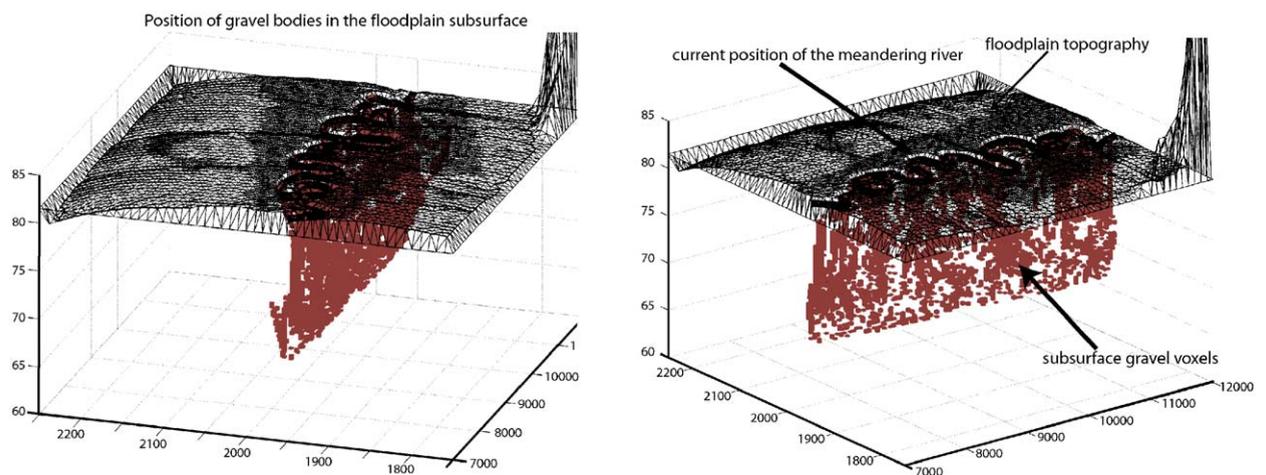


Fig. 9. Block diagrams showing distribution of paleo-channel voxels with a sand fraction larger than 0.9 in subsurface of modeled floodplain.

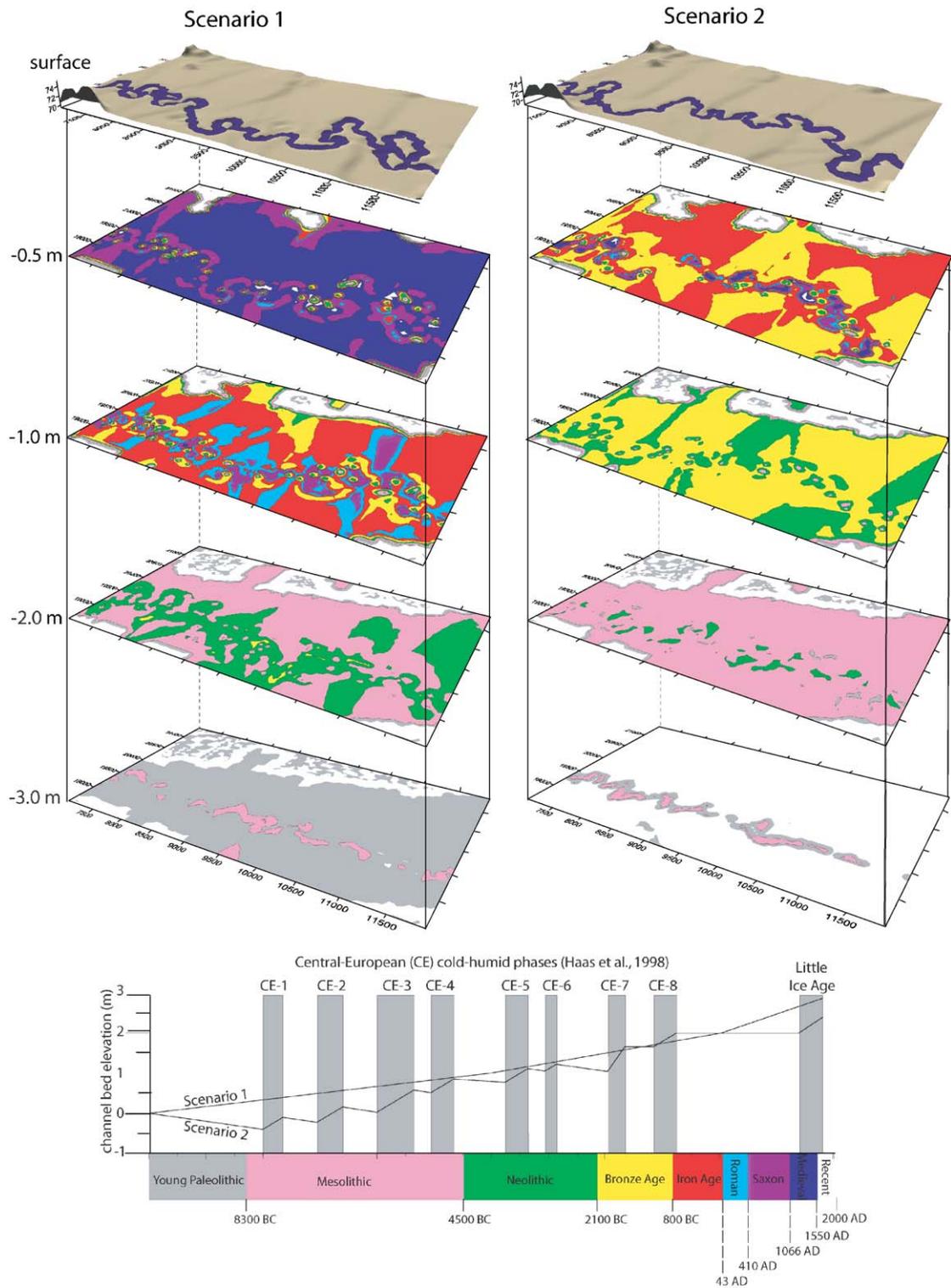


Fig. 10. Sediment ages at depths of 0.5, 1, 2 and 3 m below floodplain surface for two different scenarios, colored according to an archaeological timescale. Timing and duration of central European wet-phases was used in construction of scenarios (see text).

sediment flux to rivers (Macklin and Lewin, 2003). In the second scenario the timing and duration of the European cold-wet phases was used to drive alternating phases of channel bed aggradation and incision. High channel bed aggradation rates of 0.8 m/kyr correspond to the wet phases, whereas moderate incision is applied during dryer periods (−0.2 m/kyr), except for the Roman and post-Roman eras.

The resulting subsurface distribution of sediment ages in the subsurface of the floodplain is very different for the two scenarios. At a depth of 0.5 m, just below the plough depth, the floodplain subsurface generated by Scenario 1 is mainly of Medieval age due to the post-Roman/Medieval increase in sedimentation rate. The channel exposes older Saxon and some Roman deposits. The shallow floodplain subsurface of Scenario 2 is predominantly Bronze and Iron Age, reflecting the increase in aggradation during CE-7 and CE-8 wet phases. The central channel belt is marked by younger Saxon and Medieval deposits. The linear patches visible in the subsurfaces of both scenarios follow the undulating topography in the floodplain where swales are orientated perpendicular to the channel. Similar ridge and swale structures have been described in an archaeological context for the floodplain of the Red River, Kansas (Guccione et al., 1998). In this setting, the changing texture and archaeological resolution of floodplain sediments are explained as a function of differential floodplain sedimentation rates, controlled by the proximity and orientation of the nearby meandering channel.

5. Discussion and conclusion

An improved version of the TIN-based CHILD meander model is presented that incorporates a static regular grid for the storage of a 3D fluvial stratigraphy. A simple mapping algorithm is illustrated to transfer mass and sediment properties between the adaptive surface TIN mesh and the new grid, facilitating the creation of a stratigraphic record of TIN landscape change. The mapping algorithm is here applied in a geologic timescale simulation, but could also be used in solving “engineering time-scale” sediment transport problems on adaptive meshes with spatially varying properties of the sediment deposit.

Simulation experiments support earlier predictions that the sinuosity development of the mean-

dering channel can be described by a saw-tooth curve in which sinuosity gradually increases during periods of meander loop growth, but is punctuated by rapid falls due to the creation of cut offs (see Hooke, 2003, for an overview). The occurrence of cut-off events in the model is strikingly periodic, showing an average spacing of 1500 yr in this example. However, other sensitivity experiments, not shown here, suggest that increasing the value of key variables, such as water discharge and bank erodability, results in larger meander loops with composite bend geometry, characterized by superposed and more irregular cut off periodicities (Clevis et al., 2004b). This supports many field-based theories that state that changing climatic conditions can easily modify the basic geometry and behavior of meandering systems (Starkel, 1988; Vandenberghe, 1995, 2002; Brown et al., 2001). Further experiments, supported by dimensional analysis of the governing equations, can therefore be used to quantify these basic relationships, investigate threshold conditions, and estimate the duration of response times involved. The development of the surface meander pattern is reflected in the subsurface stratigraphy by a configuration of U-shaped paleo-channel deposits. This simulated architecture, combined with a study of the interconnectedness of the sand bodies, might be useful in modeling fluids in aquifers, oil reservoirs and the exploration of aggregates, as has been done in several other studies involving dynamic models of meander stratigraphy (Stølum and Friend, 1997; Gross and Small, 1998).

A new application of the stratigraphic simulation of meandering rivers presented here is the prediction and visualization of archaeological units in floodplain subsurface. The typical configuration of these units is complex and characterized by a patchy distribution of units along buried paleo-channels, flanked by channel-perpendicular bands of similar-aged sediment. These subsurface distributions could be compared to a study of a real floodplain using sediment auger cores and dating techniques, in order to test the algorithms and verify input variables used in scenarios. In cases where the model results seem to match observed subsurface data, these same results might be used to aid further archaeological prospecting. Alternatively, the archaeological subsurface datasets generated by the model can be used as to design and test cost-effective, non-destructive coring strategies.

Acknowledgments

Financial support for this research was provided by English Heritage as part of the Aggregates Levee Sustainability Fund, and by a Netherlands Organization for Scientific Research (NWO) fellowship to Quintijn Clevis. John C. Tipper and an anonymous reviewer are thanked for their comments, which improved the manuscript.

Appendix A. Layer interpolation routine

The layer interpolation algorithm (Tucker et al., 1999) should only be used on a static subsurface dataset, in order to avoid the progressive stratigraphic information ‘smearing’ that results from significant node movement (Fig. 1). The algorithm starts by finding the time of most recent erosion/sedimentation activity of the top layer of each of the surrounding three nodes (A). The first new layer is created from those layers that have the largest matching recent activity times (RATs). (Time is measured as length of time from the start of a simulation, therefore larger times are more recent times.) The thickness of the layer at the new node is interpolated (based on location) from the thickness of layer(s) matching RATs at each of the surrounding nodes. It is possible that one or two of the nodes will not have layers with matching RATs. For purposes of the interpolation, the algorithm treats

these layers as having zero thickness at their node positions. Other information about the layers is also interpolated, including erodibility, surface exposure time, and sediment grainsize texture. This information is a weighted average based on the thickness of the matching layers. Once an existing layer is used in the interpolation, the time of most recent activity of the next layer at that node is found. The algorithm again compares the time of most recent activity of the layers at the three different nodes and then interpolates between those that have the youngest matching RATs, and so on until the bottom layer at each node has been reached. The process of layer interpolation is most easily understood by stepping through an example. Fig. A1(B) illustrates the hypothetical alluvial layers (with their RATs) at three established nodes, along with the layers that would be formed if a new node were added between them. The interpolation algorithm first finds the RAT of the top layer at each of the three established nodes. In this case, the layer RAT of the top layer of each of the nodes is the same (200), and therefore the top layer of the new node would be assigned a RAT of 200. The thickness of the top layer at the new node would be calculated by fitting a plane in space to the three established nodes. Other attributes associated with the new layer would be averaged as explained above. The algorithm would then find the RATs of the next layers at the three surrounding nodes. In this example, node 2 has a

```

FOR each surrounding node (i=1,2,3) DO
  CurrentLayeri = top layer of node i
  Agei = recent activity time of top layer of node i
  InvDisti = reciprocal of the distance from new node to node i
ENDFOR
CurrentAge = MAX(Age1, Age2, Age3)
DO
  Normalize = 0 //used for weighting of layer properties
  Property = 0 //can be repeated for any number of layer properties
  FOR each surrounding node (i=1,2,3) DO
    IF Agei = CurrentAge DO
      Depthi = LayerDepth(CurrentLayeri)
      Property = Property + Property(CurrentLayeri)*InvDisti*Depthi
      Normalize = Normalize + InvDisti*Depthi
    IF CurrentLayer is not the bottom layer DO
      CurrentLayeri = the next lower layer at node i
      Agei = recent activity time of CurrentLayeri
    ELSE
      Agei = -1
    ENDIF
  ELSE
    depthi = 0
  ENDIF
ENDFOR
DepthofNewLayer = FitPlaneValue(depthi, x,y values of surrounding
nodes, x,y of new node)
PropertyofNewLayer = Property/Normalize
Insert new layer at the bottom of the layer list of the new node
CurrentAge = MAX(Age1, Age2, Age3)
WHILE CurrentAge > -1
  (A)

```

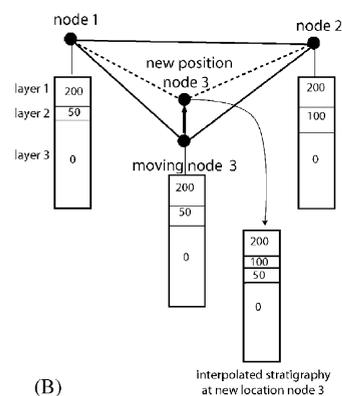


Fig. A1. Pseudo-code for interpolation of stratigraphic information of node 3 when it is translated into a new position within the TIN mesh.

larger RAT (100) than the other two nodes. Therefore, the next layer at the new node would have a RAT of 100, and its thickness would be calculated by fitting a plane to the three nodes, with node 1 and node 3 having a “z” value of zero and node 2 having a “z” value of the thickness of its second layer. The third layer at the new node would therefore have a RAT of 50. Finally, the last iteration would create the bottom layer with an RAT of 0 at the new node. An unwanted side-effect of this procedure is strong lateral diffusion of chrono-stratigraphic information as the nodes migrate.

References

- Beaumont, C., Fullsack, P., Hamilton, J., 1992. Erosional control of active compressional orogens. In: McClay, K.R. (Ed.), *Thrust Tectonics*. Chapman & Hall, London, pp. 1–18.
- Braun, J., Sambridge, M., 1997. Modelling landscape evolution on geological timescales: a new method based on irregular spatial discretisation. *Basin Research* 9, 27–52.
- Brown, A.G., Cooper, L., Salisbury, C.R., Smith, D.N., 2001. Late Holocene channel changes of the Middle Trent: channel response to a thousand-year flood record. *Geomorphology* 39, 69–82.
- Clevis, Q., de Jager, G., Nijman, W., de Boer, P.L., 2004a. Stratigraphic signatures of translation of thrust-sheet top basins over low-angle detachment faults. *Basin Research* 16 (2), 145–163.
- Clevis, Q., Tucker, G.E., Lock, G., Desitter, A., 2004b. Modelling the stratigraphy and geoarchaeology of English river systems. Report 3299 prepared for English Heritage and ALSF, 33pp. Online at <http://www.geog.ox.ac.uk/research/projects/evs/report.pdf>
- Clevis, Q., Tucker, G.E., Lancaster, S.T., Lock, G., Bras, R.L., in press. Geoarchaeological simulation of meandering river deposits and settlement distributions; a three-dimensional approach. *GeoArchaeology*, accepted.
- Densmore, A.L., Ellis, M.A., Anderson, R.S., 1998. Landsliding and the evolution of normal fault bounded mountains. *Journal of Geophysical Research* 103, 15203–15219.
- Dietrich, W.E., Smith, J.D., 1983. Influence of the point bar on flow through curved channels. *Water Resources Research* 19 (5), 1173–1192.
- Eagleson, P.S., 1978. Climate, soil, and vegetation: 2. The distribution of annual precipitation derived from observed storm sequences. *Water Resources Research* 14, 713–721.
- Gasparini, N.M., Tucker, G.E., Bras, R.L., 1999. Downstream fining through selective particle sorting in an equilibrium drainage network. *Geology* 27, 1079–1082.
- Gross, L.J., Small, M.J., 1998. River and floodplain process simulation for subsurface characterization. *Water Resources Research* 34 (9), 2365–2376.
- Guccione, M.J., Sierzchula, M.C., Lafferty, R.H., Kelley, D., 1998. Site preservation along an active meandering and avulsing river; the Red River Kansas. *GeoArchaeology* 13 (5), 475–500.
- Haas, J.N., Richoz, I., Tinner, W., Wick, L., 1998. Synchronous Holocene climatic oscillations recorded on the Swiss Plateau and at timberline in the Alps. *The Holocene* 8 (3), 301–309.
- Hooke, J., 2003. River meander behaviour and instability: a framework for analysis. *Institute of British Geographers Transactions* 28, 238–253.
- Howard, A.D., 1992. Modelling channel migration and floodplain sedimentation. In: Carling, P.A., Petts, G.E. (Eds.), *Lowland Floodplain Rivers: Geomorphological Perspectives*. Wiley, Chichester, UK, pp. 1–41.
- Howard, A.D., 1996. Modelling channel evolution and floodplain morphology. In: Anderson, M.G., Walling, D.E., Bates, P.D. (Eds.), *Floodplain Processes*. Wiley, Chichester, UK, pp. 15–62.
- Johannesson, H., Parker, G., 1989. Linear theory of river meanders. In: Ikeda, S., Parker, G. (Eds.), *River Meandering*. Water Resources Monograph 12. American Geophysical Union, Washington, DC, pp. 181–212 (Chapter 7).
- Johnson, D., Beaumont, C., 1995. Preliminary results from a platform kinematic model of orogen evolution, surface processes and the development of clastic foreland basin stratigraphy. In: Ross, G. (Ed.), *Stratigraphic Evolution of Foreland Basins*. Society for Sedimentary Geology Special Publication, vol. 52, pp. 3–24.
- Karssenbergh, D., 2002. Building dynamic spatial environmental models. Ph.D. Dissertation, Geographical Sciences, Utrecht University, Netherlands, 222pp.
- Lancaster, S.T., 1998. A nonlinear river meander model and its incorporation in a landscape evolution model. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 177pp.
- Lancaster, S.T., Bras, R.L., 2002. A simple model of river meandering and its comparison to natural channels. *Hydrological Processes* 16 (1), 1–26.
- Mackey, S.D., Bridge, J.S., 1995. Three-dimensional model of alluvial stratigraphy: theory and application. *Journal of Sedimentary Research B* 65, 7–31.
- Macklin, M.G., Lewin, J., 2003. River sediments, great floods and centennial-scale Holocene climate change. *Journal of Quaternary Science* 18, 101–105.
- Miller, S.R., Slingerland, R.L., 2003. Ramp dip and the steady-state topography of fault-bend folds. *Geological Society of America Abstracts with Programs* 35 (6), 296.
- Oualine, S., 1997. *Practical C Programming*, third ed. O'Reilly, Cambridge, MA, 428pp.
- Pizzuto, J.E., 1987. Sediment diffusion during overbank flows. *Sedimentology* 34, 301–317.
- Posamentier, H.W., 2003. Depositional elements associated with basin-floor channel-levee system: case study from the Gulf of Mexico. *Marine and Petroleum Geology* 20 (6–8), 677–690.
- Slingerland, R., Syvitski, J.M., Paola, C., 2002. Sediment modeling system enhances education and research. *EOS, Transactions AGU* 83 (49), 578–579.
- Smith, J.D., McLean, S.R., 1984. A model for flow in meandering streams. *Water Resources Research* 20 (9), 130–1315.
- Starkel, L., 1988. Tectonic, anthropogenic and climatic factors in the history of the Vistula river valley downstream of Cracow. In: Lang, G., Schluchter, C. (Eds.), *Lake, Mire and River Environments During the Last 15000 Years*. Balkema, Rotterdam, The Netherlands, pp. 161–170.
- Stølum, H.-H., 1996. River meandering as a self-organizing process. *Science* 271, 1710–1713.

- Stølum, H.-H., 1998. Planform geometry and dynamics of meandering rivers. *Geological Society of America Bulletin* 110, 1485–1498.
- Stølum, H.-H., Friend, P.F., 1997. Percolation theory applied to simulated meander belt bodies. *Earth and Planetary Science Letters* 153, 265–277.
- Sun, T., Meakin, P., Jossang, T., Schwarz, K., 1996. A simulation model for meandering rivers. *Water Resources Research* 32 (9), 2937–2954.
- Tipper, J.C., 1991. Fortran programs to construct the planar Voronoi diagram. *Computers & Geosciences* 17 (5), 597–632.
- Törnqvist, T.E., Wallinga, J., Murray, A.S., De Wolf, H., Cleveringa, P., De Gans, W., 2000. Response of the Rhine–Meuse system (west-central Netherlands) to the last Quaternary glacio-eustatic cycles: a first assessment. *Global and Planetary Change* 27, 89–111.
- Tucker, G.E., Bras, R.L., 2000. A stochastic approach to modelling the role of rainfall variability in drainage basin evolution. *Water Resources Research* 36, 1953–1964.
- Tucker, G.E., Lancaster, S.T., Gasparini, N.M., Bras, R.L., 1999. Modeling floodplain dynamics and stratigraphy: implications for geoarchaeology. Final Technical Report (Part II-C), Submitted to US Army Corps of Engineers Construction Engineering Research Laboratory (USACERL), Contract Number DACA88-95-C-0017, 16pp.
- Tucker, G.E., Lancaster, S.T., Gasparini, N.M., Bras, R.L., Rybarczyk, S., 2001a. An object-oriented framework for distributed hydrological modeling using triangular irregular networks. *Computers & Geosciences* 27 (8), 959–973.
- Tucker, G.E., Lancaster, S.T., Gasparini, N.M., Bras, R.L., 2001b. The Channel–Hillslope integrated landscape development model (CHILD). In: Doe, W.W. (Ed.), *Landscape Erosion and Sedimentation Modelling*. Kluwer, New York, NY, pp. 349–388.
- Vandenberghe, J., 1995. Timescales, climate and river development. *Quaternary Science Reviews* 14, 631–638.
- Vandenberghe, J., 2002. The relation between climate and river processes, landforms and deposits during the Quaternary. *Quaternary International* 91, 17–23.