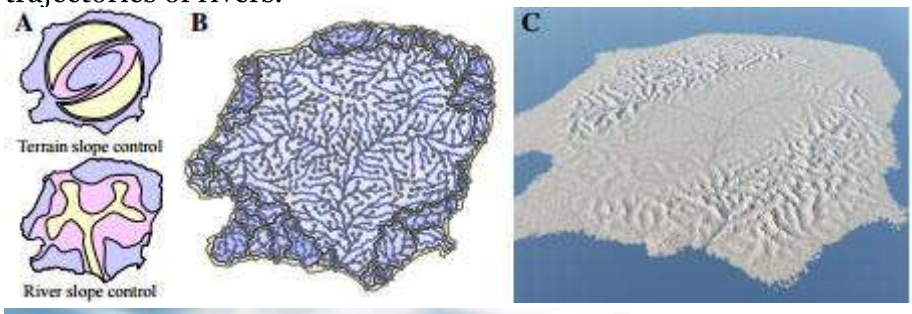
Artificial Terrain Generation

Problem: when you get very close to a terrain heightfield, it looks unrealistically flat

Synthetic Heightfield Detail

- one solution: generate a seeded random perturbation, to give the height map some high-frequency detail
- many games have used this approach, such as Joint Strike Fighter (Eidos, 1997)
- summary by *Bill Graham*:
    1. The first method is to assign real or imaginary (random) height data to a convenient lattice. Then, an iteration process takes place (using fractal Brownian noise -- fBM) to define the heightfields between the primary lattice. The iterative process is known generally as the midpoint displacement method. The roughness of the terrain is controlled by the fractal dimension (or Hurst exponent) assigned to the iterative process. Once the data points are created via iteration, one can create polygons and render a landscape.
    2. The second method is to use fBM noise to generate a matrix of phase and amplitude data (complex numbers) in the frequency domain associated with Fourier methodology, then perform an inverse Fast Fourier Transform (FFT) to create a heightfield in the time (or space) domain of the Euclidean world. The transformed matrix is rendered after conversion to an array of polygons. This method is sometimes called Spectral Synthesis. Spectral Synthesis provides a much more realistic landscape but is MUCH more computationally intensive. Furthermore, the method (at least to the limit of my understanding at this point) cannot iterate between points. The whole batch is done at once.
    3. A third set of methods are proposed in the Perlin/Musgrave book. Here, the noise is generated by the Perlin noise function which basically uses a spline function. On the surface, this approach looks very promising for sparse bathymetric data, but these guys make it clear they are not interested in the real world of empirical height (bathymetric) data and offer no examples that incorporate real data.
- Terrain Tutorial - Particle Deposition

Papers and Books

- Terrain Generation using Procedural Models based on Hydrology
    - Purdue and Universite de Lyon, Siggraph 2013
    - The "framework uses rivers as modeling elements. It first creates a hierarchical drainage network that is represented as a geometric graph over a given input domain. The network is then analyzed to construct watersheds and to characterize the different types and trajectories of rivers."
    - 

    - 
- Terrain Synthesis from Digital Elevation Models
    - Howard Zhou, Jie Sun, Greg Turk, and Jim Rehg, IEEE Viz. 2007

  - Describes a unique and powerful approach: instead of parametrically synthesizing, a heightfield is generated by assembling chunks of a real-world input.
  - 'The synthesis is guided by a user-sketched feature map that specifies where terrain features occur in the resulting synthetic terrain. The system finds patches from the example data that match the features found in the user's sketch.'
  - Examples showing the same input sketch matched to four different real-world example DEMs:
  - 
- Real-Time Editing, Synthesis, and Rendering of Infinite Landscapes on GPUs
  - Jens Schneider, Tobias Boldte, Rüdiger Westermann, 2006
  - uses "the concept of projected grids to achieve near-optimal sampling of the landscape"
  - claims to run at over 60 fps while synthesizing fractal terrain of infinite detail
  - 
- Heightfield Synthesis by Non-Parametric Sampling
  - Dachsbache, C., Meyer, M., Stamminger, M. In *Vision, Modeling and Visualization 2005*
  - paper isn't online, but is included as section 7.3 of Dachsbacher's giant paper Interactive Terrain Rendering: Towards Realism with Procedural Models and Graphics Hardware (2006)
- Infinite Fun Space (2002)
  - small academic project, closed source, procedurally generated landscapes in real time using OpenGL on Linux
  - thesis paper: Modeling for the Plausible Emulation of Large Worlds (pdf), Steven C. Dollins, BrownUniversity

- major Book: Texturing & Modeling, David Ebert, F. Kenton Musgrave
- J. P. Lewis, Generalized stochastic subdivision (pdf). *ACM Transactions on Graphics,* July 1987
- A. Fournier, D. Fussel, and L. Carpenter. Computer rendering of stochastic models, *Communications of the ACM,* 1982

Physically-based Detail Generation

- **erosion**
  - another possibility would be to simulate processes such as **erosion** to produce believable detail
  - ideally, you would start from a real elevation map (eg. 10-30 m/pixel) and a map of soil type, then use erosion to produce the high-frequency detail (eg. 1 m/pixel)
  - see C. Burke's pages on Generating Terrain and Erosion and Hydrology
  - see Hydrogeology on this site
  - *Terrain simulation using a model of stream erosion*, Kelley, Malin, and Nielson, Comm of ACM. 1988, pp. 263-268

- **plate tectonics**
  - paper, Physically Based Terrain Generation : Procedural Heightmap Generation Using Plate Tectonics (Lauri Viitanen 2012) and video: Real Time Plate Tectonics Simulation

Rocks

- it can help realism immensely to augment the height field with polygonal 3D rocks, embedded in the ground
- ideally a randomized procedural generator would produce the rocks
- MacRock/PCRock is a old small free utility which has parameters for complexity and smoothness, outputs DXF
- Mark Stock's Rocktools is a command-line tool that can create and modify random rocks, outputs OBJ
- rock generation is a feature of commercial landscape programs (AWB, WCS, etc.) and a few modeler plugins
- from Stephen Ervin's Digital Landscape Modeling and Visualization:
  - "One problem in the landform category is the simple task of modeling a rock - whether a boulder, rock outcropping, or basic constituent of a dry laid stone wall. These humble objects have geometries and surface detail not easily represented in any 3D modeling system, and simplified versions of them always seem just that -- oversimplified. Photographic texture mapping , combined with bump -mapping on suitably formed 3D solids has been used for some rock models, to good visual effect."
- Dachsbacher's giant paper Interactive Terrain Rendering: Towards Realism with Procedural Models and Graphics Hardware (2006) has sections on generating and GPU-rendering of procedural rocks

Synthetic Terrain - Articles

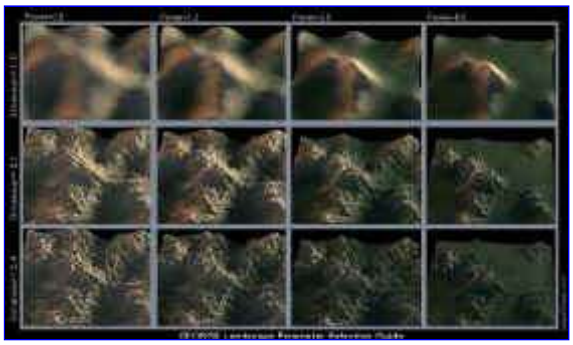
- Generating Planetary Bodies by Sean O'Neil, on Gamasutra
  - discusses Perlin Noise and fBM
  - uses the 3D surface point as the input to the noise algorithm, so it doesn't suffer from bunching at the poles, as it would if it just stretched 2D noise onto a sphere
  - includes demo with OpenGL source
  - as of 2005, all of Sean's work is rolled up on his site A Real-Time Procedural Universe which includes demos with source for procedural planets, atmospheric scattering, cloud and grass rendering experiments:


- Spherical Landscapes by Hugo Elias
  - describes one way of creating undistorted continents on planets, using repeated slicing planes through the sphere
  - independent of how the planet's mesh is specified
- Modelling fake planets by Paul Bourke
  - similar to Hugo Elias's approach, with simple source code example
- Texture-synthesis for planetary-scale terrains is part of Dave Hill's research
- Generating Random Fractal Terrain by Paul Martz
- Using Bitmaps for Automatic Generation of Large-Scale Terrain Models by Kai Martin

Synthetic Terrain - Applications

- for many applications that deal with synthetic terrain, see see **Artificial Terrain Tools & Software Packages**
- *Procedural Planet's tech notes*
  - articles on generating massive fractal texture/heightfield and using SIMD
  - there is a corresponding Win32 demo program (no source)

- John Beale
  - author of heightfield generators Gforge and HF-lab
  - Gforge is a command-line utility to generate a heightfield using an FFT algorithm
  - also has utilities to simulate erosion
  - Unix sources, and MSDOS executables only
  - offline generation only, he uses POV-Ray for rendering



- terraineer is a small open-source utility which can be used to generate and to learn to generate heightmaps, including a handle of random algorithms
- many MMOG (massively multiplayer online games) face the problem of generating content (fantasy worlds), and many of them attempt to solve this problem with procedural terrain generation
  - one example is Infinity. "**Procedural universe:** procedural programming is a technique to let the computer generate the game universe on-the-fly, in real-time on request, rather than manually building everything. Because of this, the generated universe can be absolutely huge. In Infinity, billions of worlds, most of them never explored by any player, are awaiting the adventurous soul."