This tutorial shows several ways to create animated waters with ocean shaders in Maya and in Virtools.

1- Ocean shaders in Maya

Let's create an ocean, go to Dynamics, Fluid Effects > Ocean > Create Ocean.

Zoom out to see the ocean plane. You need to add a preview plane in order to preview the simulation.

To preview press the double arrow and press play.

Render the ocean. Go to Fluid Effects > Ocean > Add Preview Plane.

Let's create a light Create > Light> Directional Light. The light is created by default in the center of the grid. Select the light, move the light above the ocean.

Go to the attribute editor change intensity = 2.

Rotate the light: –100 42 20

Let's modify the attributes for the ocean shader

Select the review plane or the ocean plane. Go to the Attribute Editor, select the oceanShader1 tab.

Open the Ocean attributes section and change

Num Frequencies = 12. Frequencies add texture to the water surface.

Wave Length Max = 40 Wave length controls the size of the waves and their vertical amplitude.

Go to the start of playback range. Play the simulation. Render.
Let's add wave crests on the top of the waves. Wave peaking simulates side-to-side drifting of waves. Click on the wave peaking graph in order to add markers. Render. Go to specualr shading Specularity = 0.45 creates brighter highlights Eccentricity = 0.1 creates bigger highlights

Let's create a boat. Create a polygonal cube. Go to Modeling > Create > Polygonal Primitives > Cube. In the Attribute Editor, select pCube1 and scale: 10 10 10

Shift select the cube and the ocean plane - NOT the preview plane
Go to Fluid Effects > Ocean > Make Boats
Please note that Maya creates a boat locator at the center of the cube to mark the cube's position in space. Play the simulation the cube moves like a boat in the ocean. Let's change the buoyancy. Select the Locator – that's easier in Outliner - Go to the attribute Editor > locatorShape2 tab > Extra Attributes > Buoyancy = 0.75, Pitch = 0.2, Roll = 0.1

Let's create a Motor Boat. Select the cube-boat, go to Dynamics, Fluid Effects > Oceans > Make Motor Boat. Set Throttle and Rudder values for the boat. Create several keyframe for the boat's locator crossing the ocean. Play back the animation. Render.
Let's create a boat wake. Select the cube-boat, go to Dynamics, Fluid Effects > Oceans > Create Wakes. This creates a fluid emitter in order to drive the motion of the fluid. Play back the animation. Render.

**2- Time based waves in Maya and in Virtools**

Using Maya Hypershade > Utilities and Virtools > Schematics, you can compare how similar time based behaviors create deformations of mesh and textures of ocean waves in Maya and Virtools. Maya uses a timer, Virtools uses a Bezier Curve to control time.
Noise controls the mesh deformation and Texture Sine controls caustics, a way to scroll a texture on the mesh.

3- Interactive ocean shaders in Virtools

A- Simple ocean shader

This Simple ocean shader with animated bump map and geometric waves is based partly on an example from the section "Effective Water Simulation From Physical Models" from the book "GPU Gems".

Open Virtools, go to File > load OceanShader.cmo
Please note that this file needs external files called waves2.dds and CloudyHillsCubemap2.dds in order to work properly.
Select the object, RMB and choose Material Setup

In Material Setup, go to Edit Shader - pull down menu, choose "Ocean Water". Go to Shaders Parameters. Edit the parameters the same way you did for the Maya ocean.
Go to Editor > Shader Editor, choose Ocean Shader, open the code

```csharp
float4x4 worldMatrix : World; < string UIWidget = "none"; >;             // World or Model
matrix float4x4 wvpMatrix : WorldViewProjection; < string UIWidget = "none"; >; // Model*View*Projection
float4x4 worldViewMatrix : WorldView; < string UIWidget = "none"; >;
float4x4 viewInverseMatrix : ViewInverse; < string UIWidget = "none"; >;
float time : Time; < string UIWidget = "none"; >;

THIS SECTION LOADS EXTERNAL FILES NEEDED FOR OCEAN SHADER

texture normalMap
    <
        string ResourceName = "waves2.dds";
        string ResourceType = "2D";
    >;
texture cubeMap
    <
        string ResourceName = "CloudyHillsCubemap2.dds";
        string ResourceType = "Cube";
    >;
```

CODE EDITOR AND COMPILED

```csharp
// float4x4 waterColor = lerp(deepColor, shallowColor, tacing);
return waterColor*waterAncient + reflection*reflectionColor*water;

// return waterColor;  
// return Fresnel;  
// return reflection; 

code editor and compiler

//


//

texture PS2D
    <
        string ResourceName = "waves2.dds";
        string ResourceType = "2D"
    >;
texture PS2D
    <
        string ResourceName = "CloudyHillsCubemap2.dds";
        string ResourceType = "Cube"
    >;
```
sampler2D normalMapSampler = sampler_state  
{ 
    Texture = <normalMap>;
    #if 0
        // this is a trick from Halo - use point sampling for sparkles
        MagFilter = Linear;
        MinFilter = Point;
        MipFilter = None;
    #else
        MagFilter = Linear;
        MinFilter = Linear;
        MipFilter = Linear;
    #endif
}; 
samplerCUBE envMapSampler = sampler_state  
{ 
    Texture = <cubeMap>;
    MinFilter = Linear;
    MagFilter = Linear;
    MipFilter = Linear;
    AddressU = Clamp;
    AddressV = Clamp;
};

THE FOLLOWING PARAMETERS ARE SLIDERS AVAILABLE IN MATERIALS SETUP >

SHADER PARAMETERS

float bumpHeight
<
    string UIWidget = "slider";
    float UIMin = 0.0; float UIMax = 2.0; float UIStep = 0.01;
    string UIName = "Bump Height";
>  = 0.1;

float2 textureScale
<
    string UIName = "Texture scale";
>  = { 8.0, 4.0 };

float2 bumpSpeed
<
    string UIName = "Bumpmap translation speed";
>  = { -0.05, 0.0 };

float fresnelBias
<
    string UIName = "Fresnel bias";
    string UIWidget = "slider";
    float UIMin = 0.0; float UIMax = 1.0; float UIStep = 0.01;
>  = 0.1;

float fresnelPower
<
    string UIName = "Fresnel exponent";
    string UIWidget = "slider";
    float UIMin = 1.0; float UIMax = 10.0; float UIStep = 0.01;
>  = 4.0;

float hdrMultiplier
<
    string UIName = "HDR multiplier";
    string UIWidget = "slider";
    float UIMin = 0.0; float UIMax = 100.0; float UIStep = 0.01;
>  = 3.0;

float4 deepColor
<
    string UIName = "Deep water color";
>  = {0.0f, 0.0f, 0.1f, 1.0f};

float4 shallowColor
<
    string UIName = "Shallow water color";
>  = {0.0f, 0.5f, 0.5f, 1.0f};

float4 reflectionColor : Specular
<
    string UIName = "Reflection color";
>  = {1.0f, 1.0f, 1.0f, 1.0f};

// these are redundant, but makes the ui easier:
float reflectionAmount
<
    string UIName = "Reflection amount";
    string UIWidget = "slider";
float UIMin = 0.0; float UIMax = 2.0; float UIStep = 0.01;
> = 1.0f;

float waterAmount
<
  string UIName = "Water color amount";
  string UIWidget = "slider";
  float UIMin = 0.0; float UIMax = 2.0; float UIStep = 0.01;
> = 1.0f;

float waveAmp
<
  string UIName = "Wave amplitude";
  string UIWidget = "slider";
  float UIMin = 0.0; float UIMax = 10.0; float UIStep = 0.1;
> = 1.0;

float waveFreq
<
  string UIName = "Wave frequency";
  string UIWidget = "slider";
  float UIMin = 0.0; float UIMax = 1.0; float UIStep = 0.001;
> = 0.1;

struct a2v {
  float4 Position : POSITION;   // in object space
  float2 TexCoord : TEXCOORD0;
  float3 Tangent  : TEXCOORD1;
  float3 Binormal : TEXCOORD2;
  float3 Normal   : NORMAL;
};

struct v2f {
  float4 Position  : POSITION;  // in clip space
};

B- Advanced water shader

Water shader combines waves, reflections above water and under water. For example, look at the detail of the boat. A- the water texture has continuous waves, B- the texture of the sky is reflected in the surface of the water and added to the water texture, C- the texture of an object under water is also added to the water texture. The script for Water shader combines several building blocks including Water1.1, a shader used for the rendering of materials and textures and the other steps for rendering reflections in the water. Download watershaderboat.cmo
4- Interactive ripple textures
Ripples and dust following a character’s foot steps can be very expressive ways “to make believe”. The following examples show how to create interactive ripples that can follow a character or be triggered by particles falling on a plane
Download watertexture.cmo
This interactive texture follows mouse movements on the screen. We are going to reuse this texture for a character’s footsteps. Download rainscene.cmo

The Animation Synchronizer BB can activate messages – “right foot”, “left foot” following the character’s footsteps. See following illustration.
This script attached to the character shows how the textures are located under the footsteps and under the particle rain. Please note the 3D ripple effects created by the displacement mesh.
mesh displacement using the Rain Texture