

```

void Reactor::AdjustModelMesh( Event* event )
{
    StartTiming( "Reactor::AdjustModelMesh" );
    const Hammer* hammer = event->GetHammer();
    if ( !hammer || !hammer->IsMHL() ) return;
    Point3dDouble direction = event->GetDirection();
    m_model->AdjustModelMesh( direction, hammer->GetFeaturePoints() );
    EndTiming();
}

```

$$\int_{-a}^{+a} \frac{(2^x + 4^x)}{2} dx$$

$$\begin{pmatrix} H \\ H' \\ H'' \end{pmatrix} = \begin{pmatrix} \Omega^{-2}(1 - \cos\Omega T) \\ \Omega^{-1}\sin\Omega T \\ \cos\Omega T \end{pmatrix}$$

$$\approx \begin{pmatrix} \frac{T^2}{2} \\ T \\ 1 \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{T} \\ \frac{1}{T^2} \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

```

IntersectionInfo** Reactor::GetSortedIntersections( const Event* event )
{
    Reactor::Point3dDouble direction = event->GetDirection();
    bool m_hammer;
    int m_vertices;
}

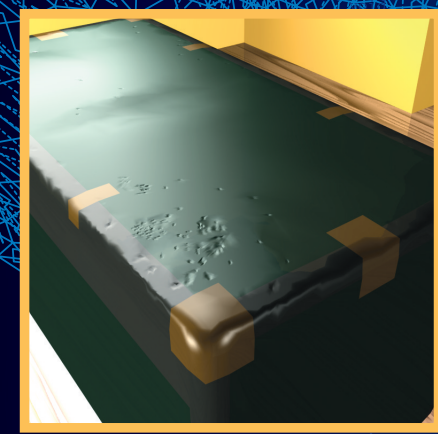
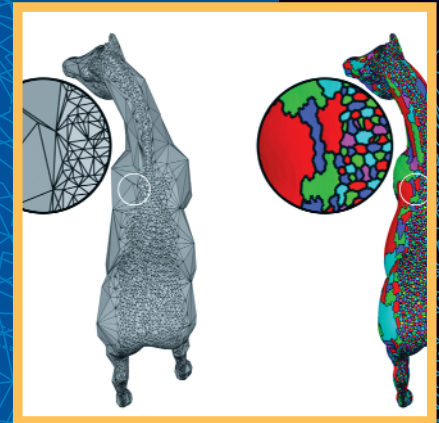
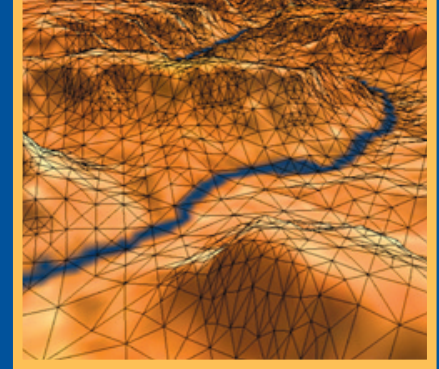
```

$$\int_{-\infty}^{+\infty} \frac{(2^x + 4^x)}{2} dx$$

$$R^N = M^{-1}R^Y$$

$$V^N = M^{-1}V^Y$$

Graphics Interface 2001



Proceedings
Graphics Interface 2001

7-9 June 2001
Ottawa, Ontario
Canadian Human-Computer
Communications Society



```

void Reactor::AdjustModelMesh( Event* event )
{
    StartTiming( "Reactor::AdjustModelMesh" );
    const Hammer* hammer = event->GetHammer();
    if ( !hammer || !hammer->IsMHL() ) return;
    Point3dDouble direction = event->GetDirection();
    m_model->AdjustModelMesh( direction, hammer->GetFeaturePoints() );
    EndTiming();
}

```

www.graphicsinterface.org

ISSN 0713-5424
ISBN 0-9688808-0-0



```

Stats:m_curPro = EndTiming();

void Reactor::GetModelBeamFacesCopy( ... )
{
    StartTiming( "Reactor::GetModelBeamFacesCopy" );
    for ( list<Face*>::iterator it = m_copyBeamFaces.begin(); it != m_copyBeamFaces.end(); ++it )
        delete *it;
    m_copyBeamFaces.clear();
    for ( vector<Vertex*>::iterator it = m_copyBeamVertices.begin(); it != m_copyBeamVertices.end(); ++it )
        delete *it;
    m_copyBeamVertices.clear();
    m_model->GetCopyOfFacesAndVertices( m_copyBeamFaces, m_copyBeamVertices );
}

```

