PHYSX 4: RAISING THE FIDELITY AND PERFORMANCE OF PHYSICS SIMULATION IN GAMES

Kier Storey, Principal Software Engineer | Date | Booth S466| South Hall

PHYSX

Scalable, robust rigid body simulation

Cross-platform

Multi-threaded and GPU accelerated

Visual debugging and profiling tools

Integrated into numerous tools and existing game engines

RIGID BODIES

Efficient and accurate distance-based contact generation

Sweep-based CCD and speculative CCD

Support for common joint types (spherical, revolute, fixed, prismatic, distance, custom)

Fast and robust iterative solver

Efficient scene query system

Flexible filtering system

Extensions for vehicles and characters

Two interfaces: “retained” and “immediate” mode

PHYSX 4

Open-source BSD license

TGS: New, more robust, non-linear rigid body solver

New reduced coordinate articulation implementation

Provides game-level performance with robotics-quality simulation fidelity

More scalable broad phase

Overhauled joint implementation

Optimizations for complex multi-shape actors

TGS SOLVER

Non-linear iterative solver

Improved robustness to ill-conditioned cases

Significantly faster converging than PGS

Improved high-mass ratio simulation

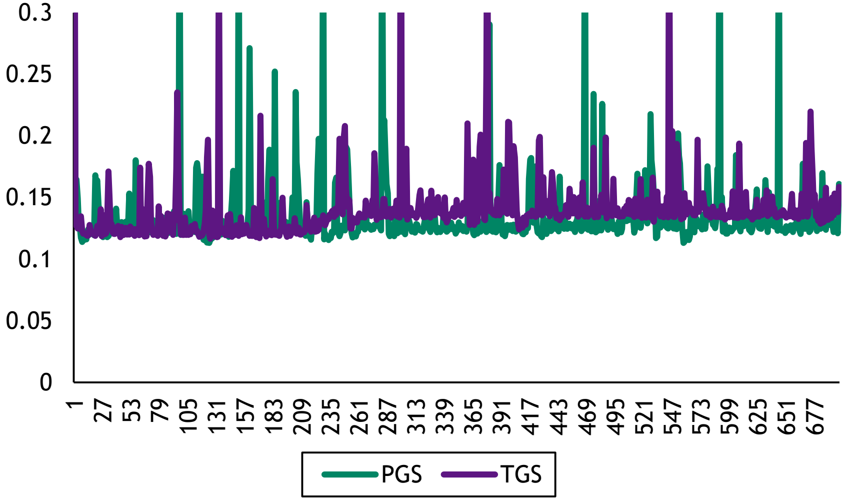
Provides “gentle de-penetration” without damaging convergence

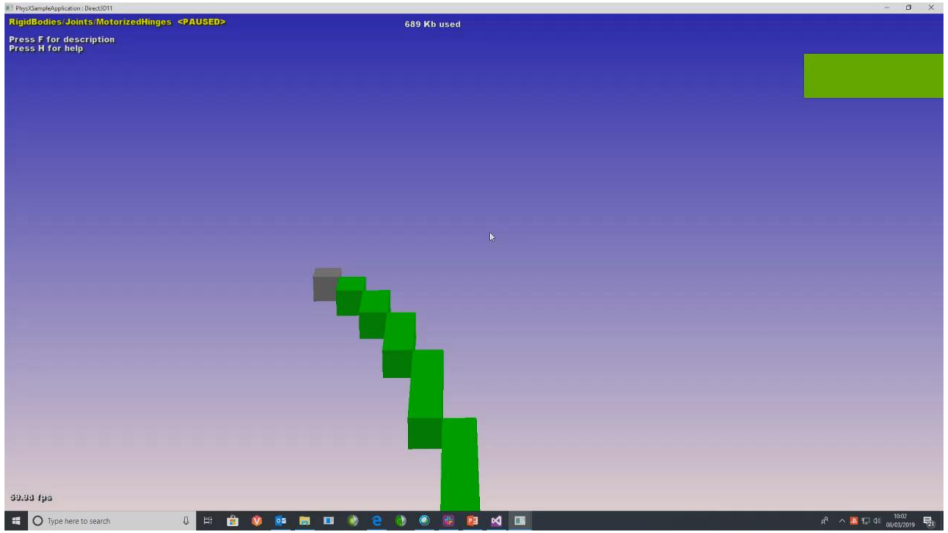
Improved joint drive handling

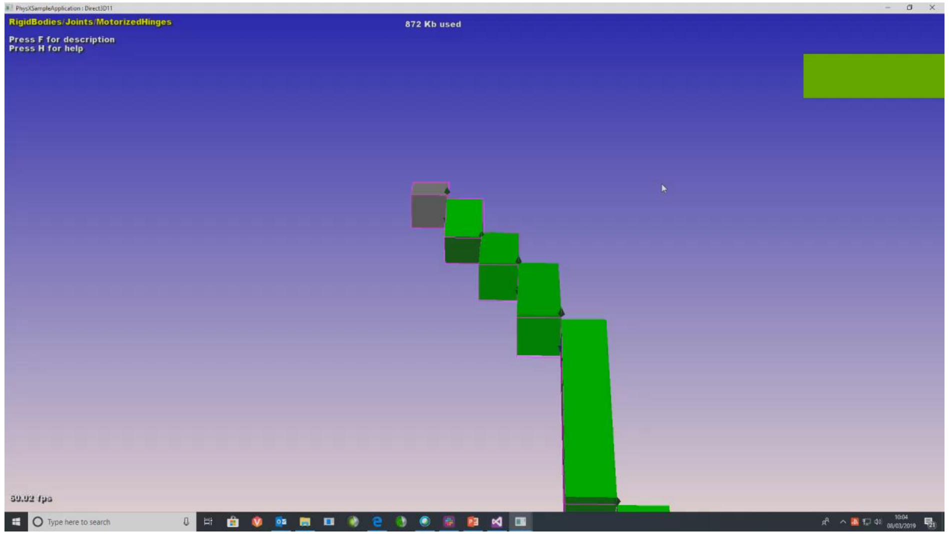
Similar performance to PGS solver

TGS COMPARISON

Revolute Chains with Drives

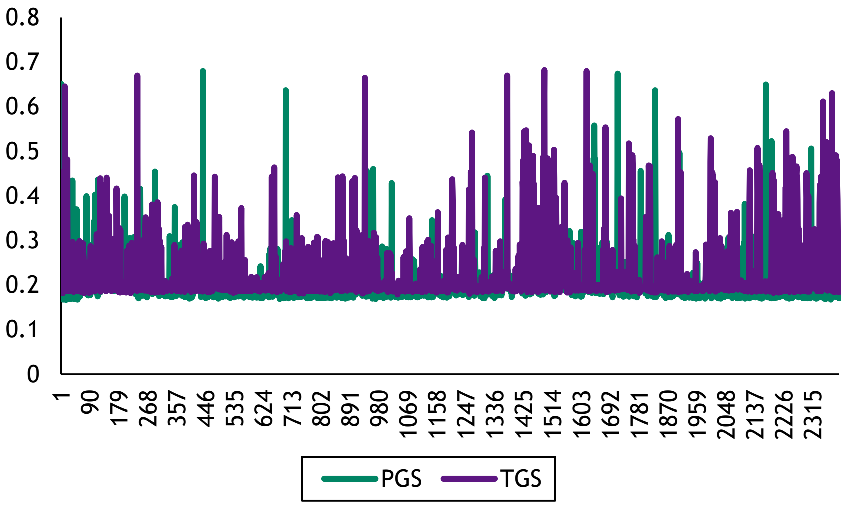


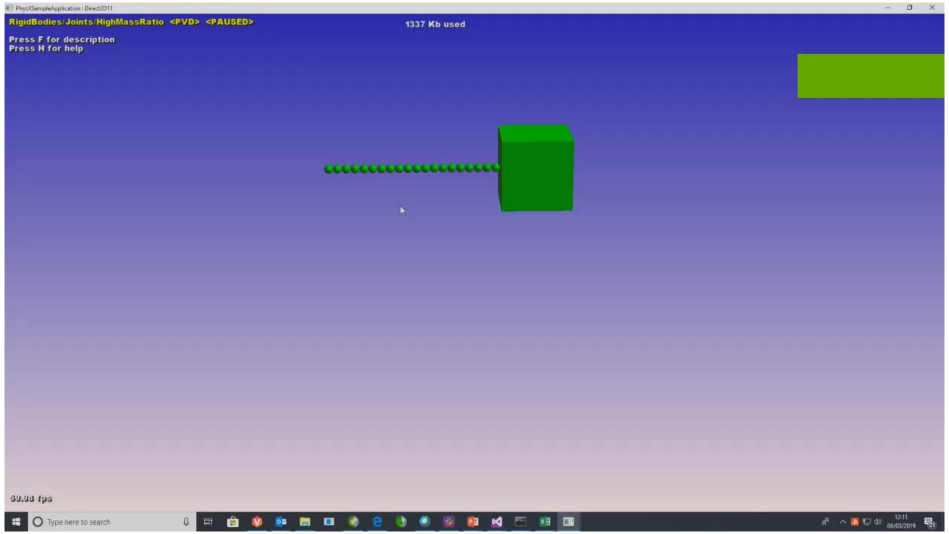


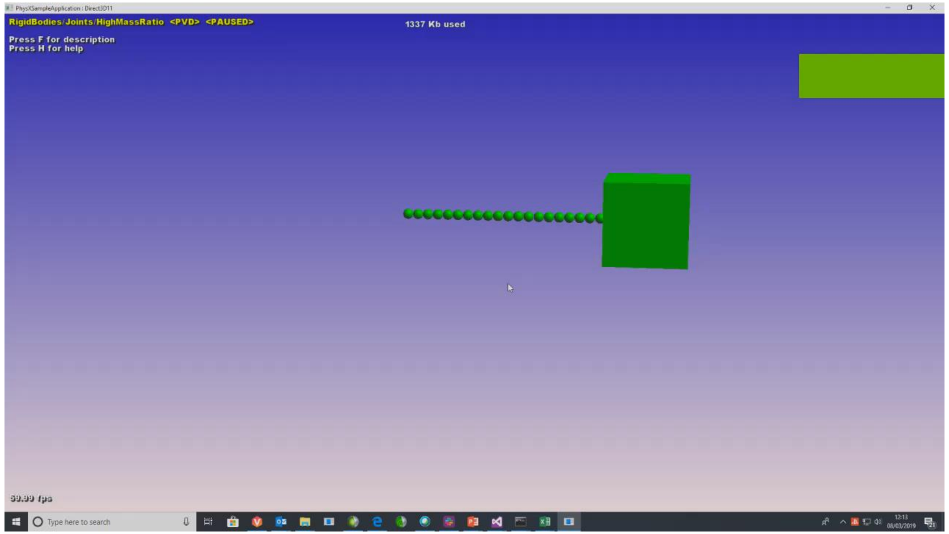


TGS COMPARISON

High-mass Ratio Chains

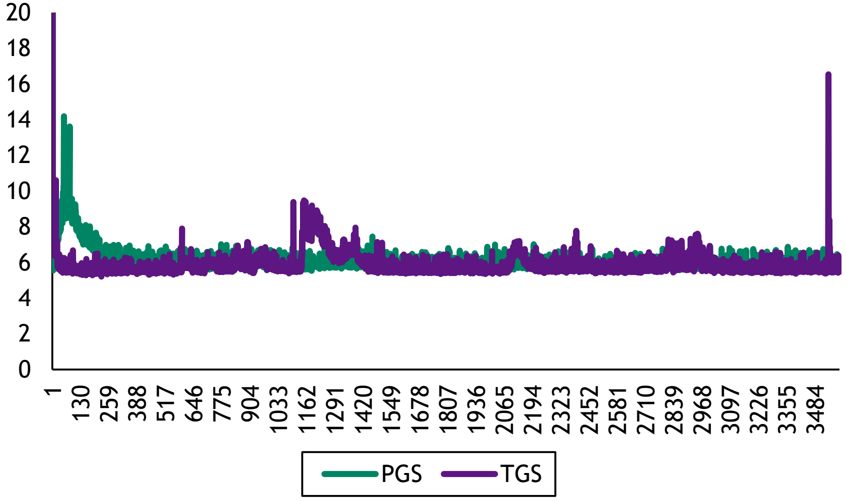


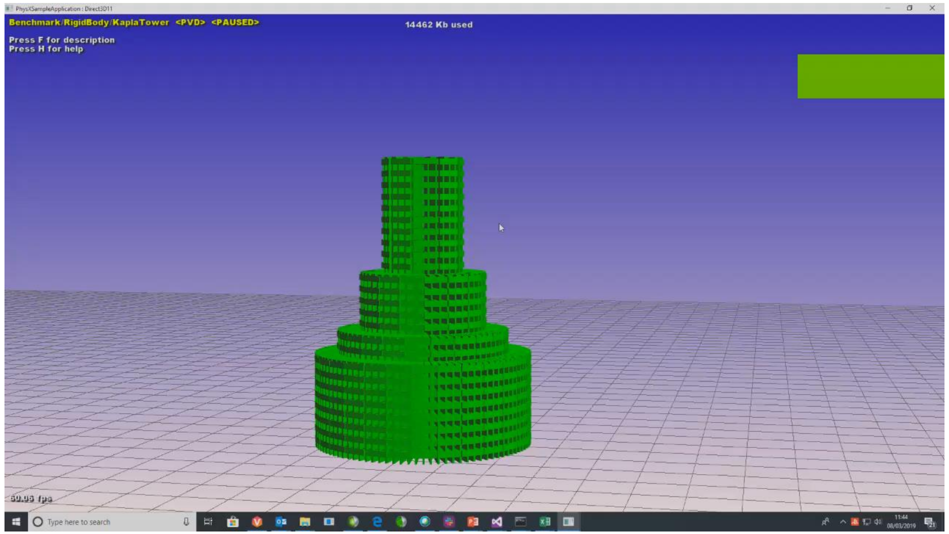


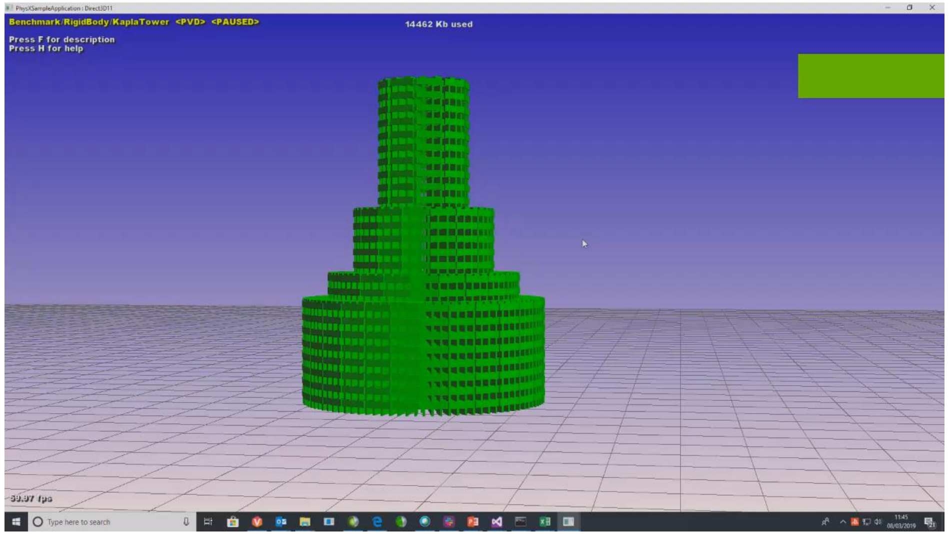


TGS COMPARISON

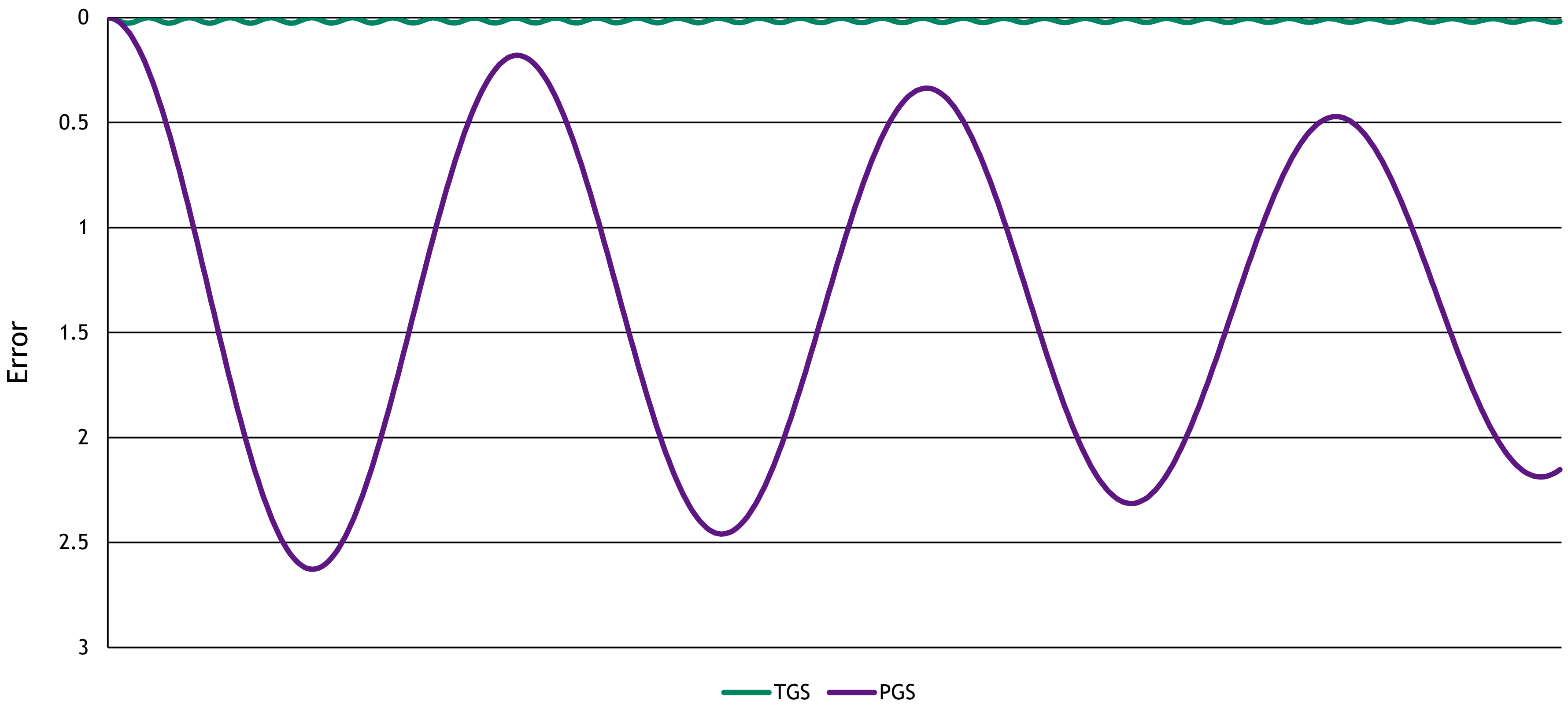
Toy-scale Kapla tower







TGS CONVERGENCE



REDUCED COORDINATE ARTICULATIONS

Hierarchical kinematic tree multi-body system

Currently supports fixed, revolute, spherical and prismatic joints

Support for limits, PD drive controllers and joint friction

Loop joints, self-collisions and interactions with external bodies handled through PGS/TGS solver

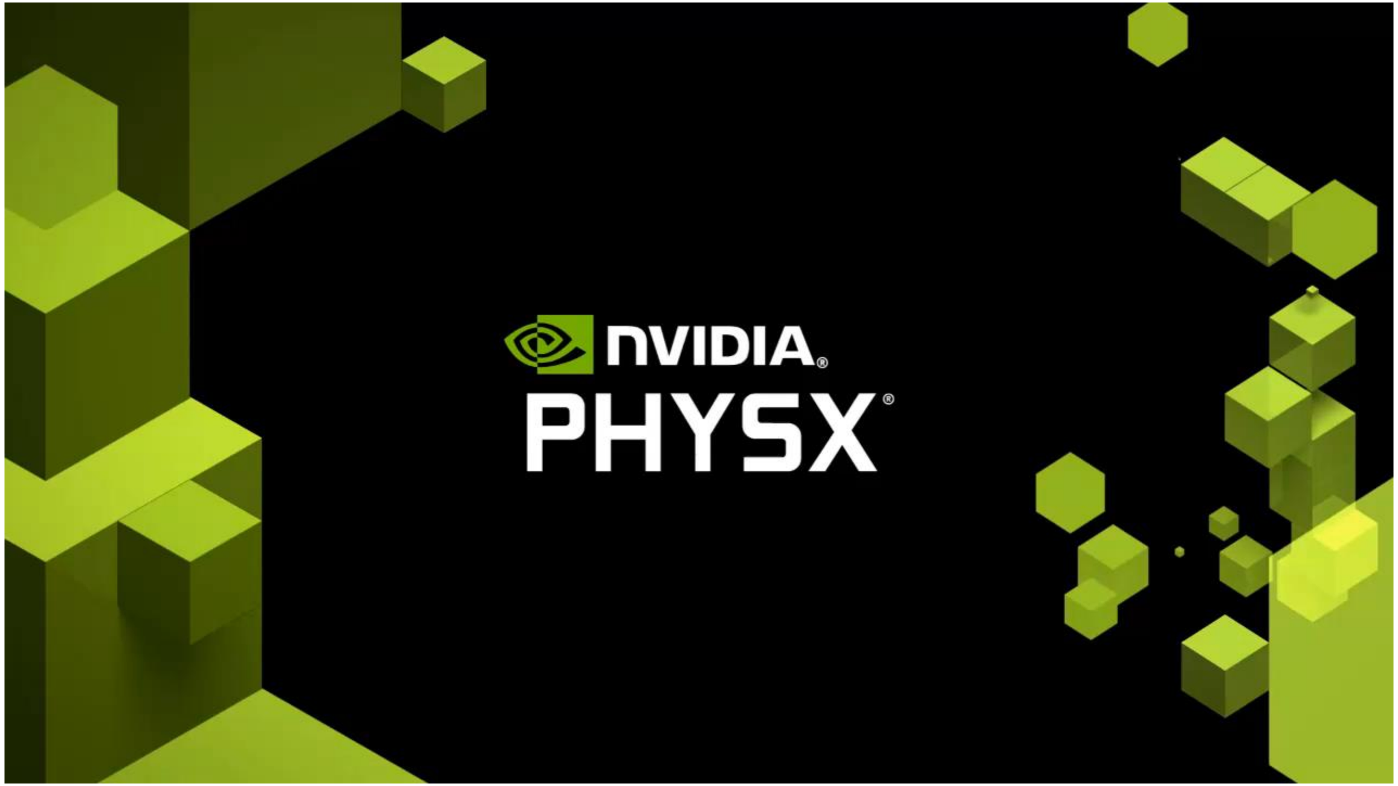
Inverse dynamics

Closely-matches analytical models

Deep integration with TGS solver yields improved simulation robustness

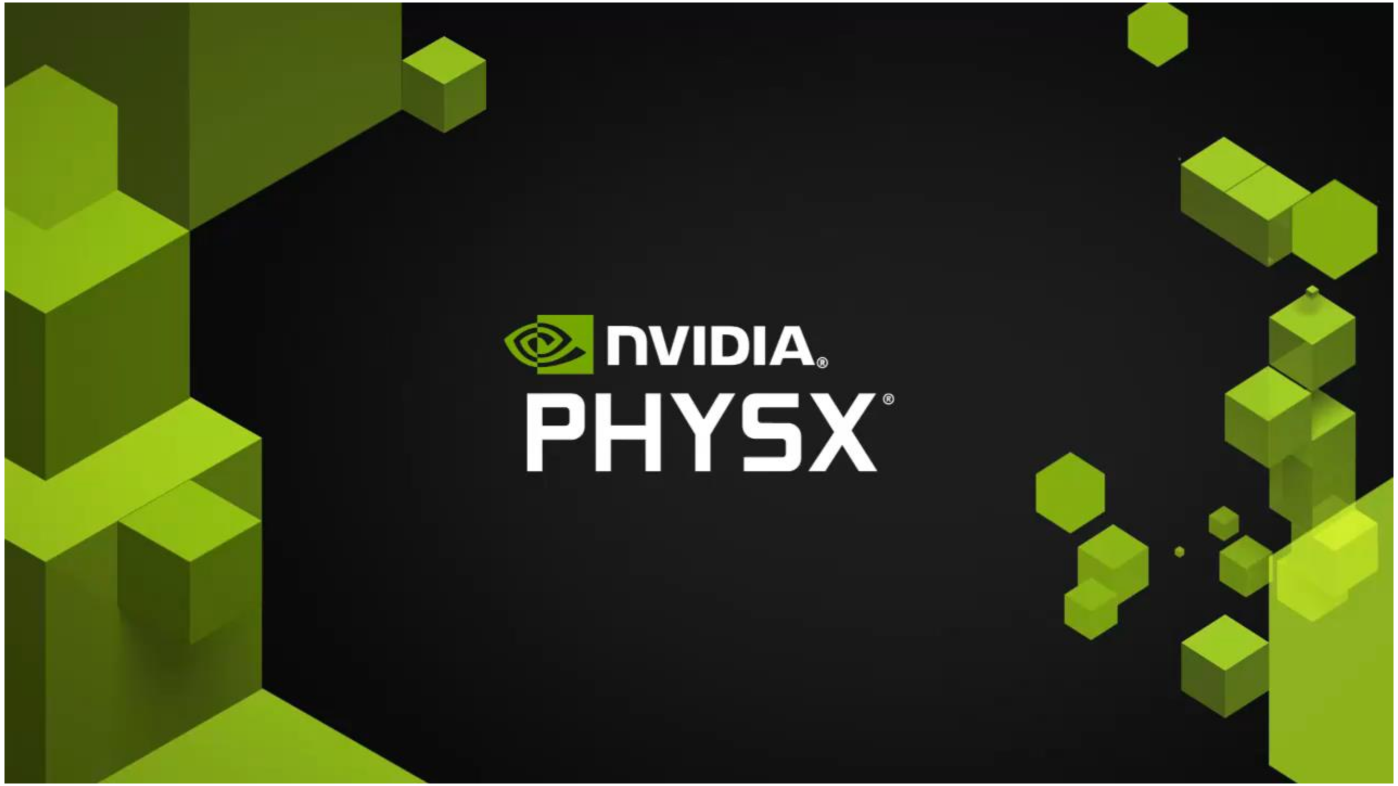
RESULTS

PhysX Warehouse Demo



RESULTS

ANYmal training



IMMEDIATE MODE

Direct access to PhysX low-level functionality

Contact generation

Constraint definitions (contacts and joints)

Low-level rigid body solver

Extremely low-overhead. Ideally-suited for simulating small sets of bodies with as little latency as possible.

RETAINED MODE

```
PxScene* scene = gPhysics->createScene(desc);

scene->addActor(*PxCreatePlane(*gPhysics,
      PxPlane(0,1,0,0), material);

PxShape* box = gPhysics->createShape
(PxBoxGeometry(PxVec3(1.f), material);
PxRigidDyn* dyn;

for(int i = 0; i < 1000; ++i)
{
      dyn = gPhysics->createRigidDynamic
      (PxTransform(PxVec3(0.f, 1.f+2.f*i, 0.f));
      dyn->attachShape(*box);
      updateMassAndInertia(*dyn, 1.f);
      scene->addActor(*dyn);
}

for(int frame = 0; frame < 100; ++ frame)
{
      scene->simulate(1.f/60.f);
      scene->fetchResults(true);
}
```

IMMEDIATE MODE

```
using namespace physx::immediate;
const int nbBodies = 10;
PxRigidBodyData bodyDescs[nbBodies];
PxSolverBodyData data[nbBodies+1];
PxSolverBody solverBodies[nbBodies+1];
PxConstraintDesc contacts[nbBodies];
PxConstructStaticSolverBody(
PxTransformFromPlaneEquation(PxPlane(0,1,0,0)),
data[0]);
PxConstructSolverBodies(bodyDescs, &data[1], 10);
PxReal dt = 1.f/60.f, invDt = 60.f;

for(int frame = 0; frame < 100; ++frame)
{
      applyGravity(&data[1],gravity,nbBodies,dt);
      int cts = 0;
      for(int i = 0; i < nbBodies; ++i){
            if(generateContacts(boxGeom, planeGeom,
                  data[i+1],data[0],contacts[cts]))
                  cts++;

      for(int i = 0; i < cts; ++i)
            createConstraints(contacts[i], invDt);

      solveConstraints(contacts, solverBodies);
      integrateSolverBodies(solverBodies);
}
```

GPU RIGID BODIES

Hybrid CPU/GPU rigid body simulation

Enabled with flags on the scene

Exact same interface and feature set as CPU simulation

Highly-scalable GPU broad phase

GPU-accelerated distance-based contact generation

GPU-accelerated PGS and TGS solver

Automatically handles transfer of data between host and device

Designed for interactive/gameplay-effecting simulation, not just visual effects

NVIDIA

PHYSX 4.1

Michelle Lu, Principal Software Engineer | 14 March, 2019 | Booth S466| South Hall

www.nvidia.com/GDC

PHYSX 4.1

Extends immediate mode to support articulations and TGS solver

GPU-accelerated articulations (coming soon)

Significantly optimized CPU articulations

Optimizations and bug fixes

ARTICULATION SOLVER

Forward dynamics optimizations and fixes

Optimized contact/joint solver solves simple constraints (articulation-rigid) in O(1) time rather than O(logn) time

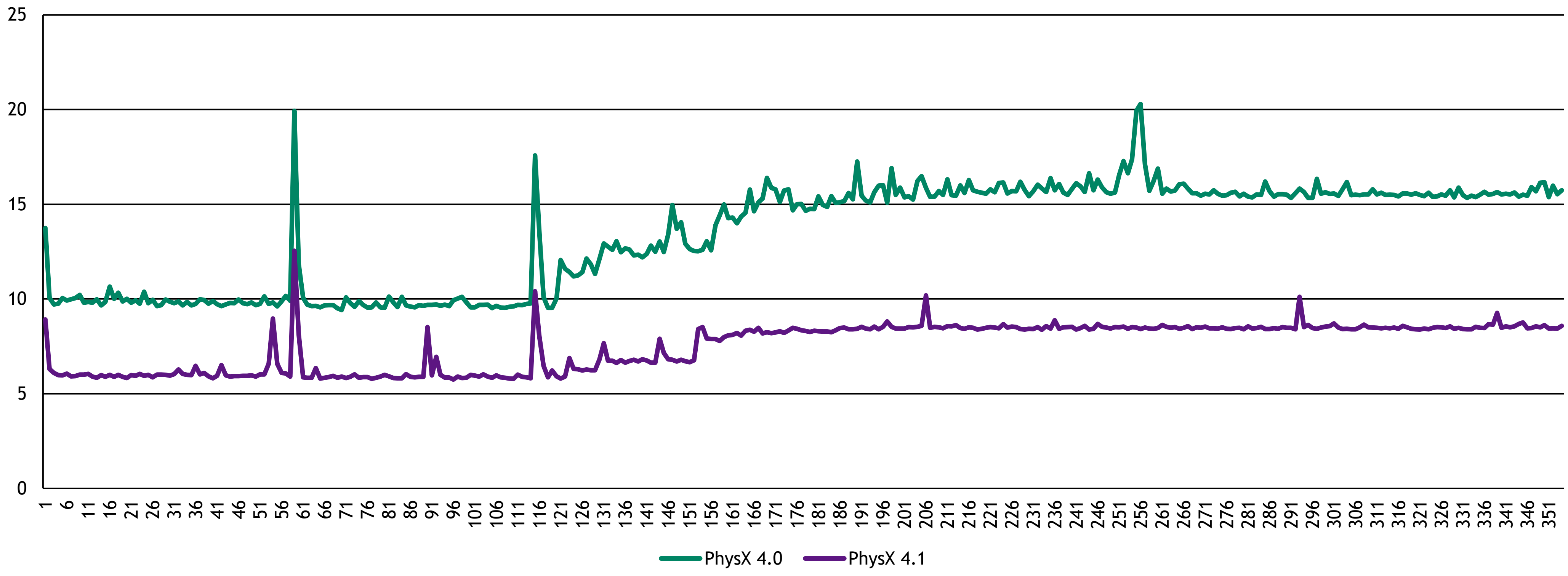
Improved accuracy of joint drives/limits

Improved convergence

Improved spherical joint support

PERFORMANCE IMPROVEMENTS

1000 24 DOF Articulations on the ground



IMMEDIATE MODE ARTICULATIONS

Small/simple API

Direct access to link/joint properties

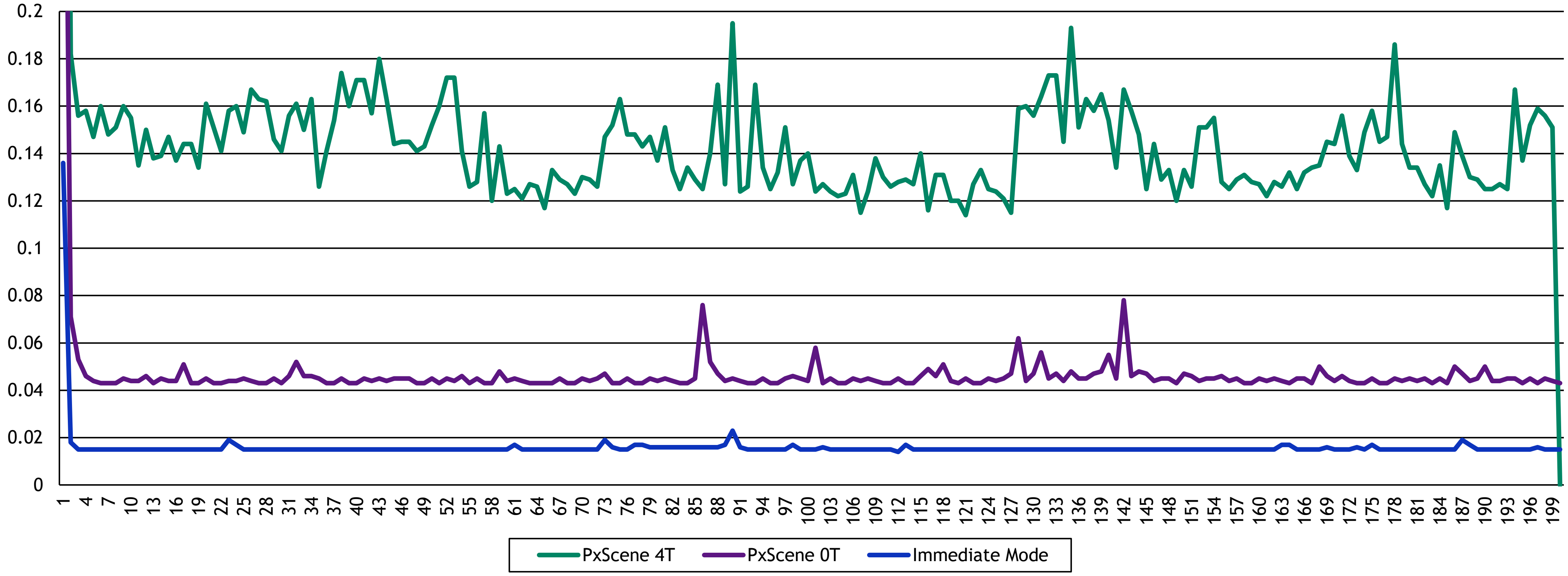
Direct simulation functions for forward dynamics on a single articulation

Coupled solver function that solves constraints between articulations and rigid bodies

New snippets showcasing immediate mode articulations and joints

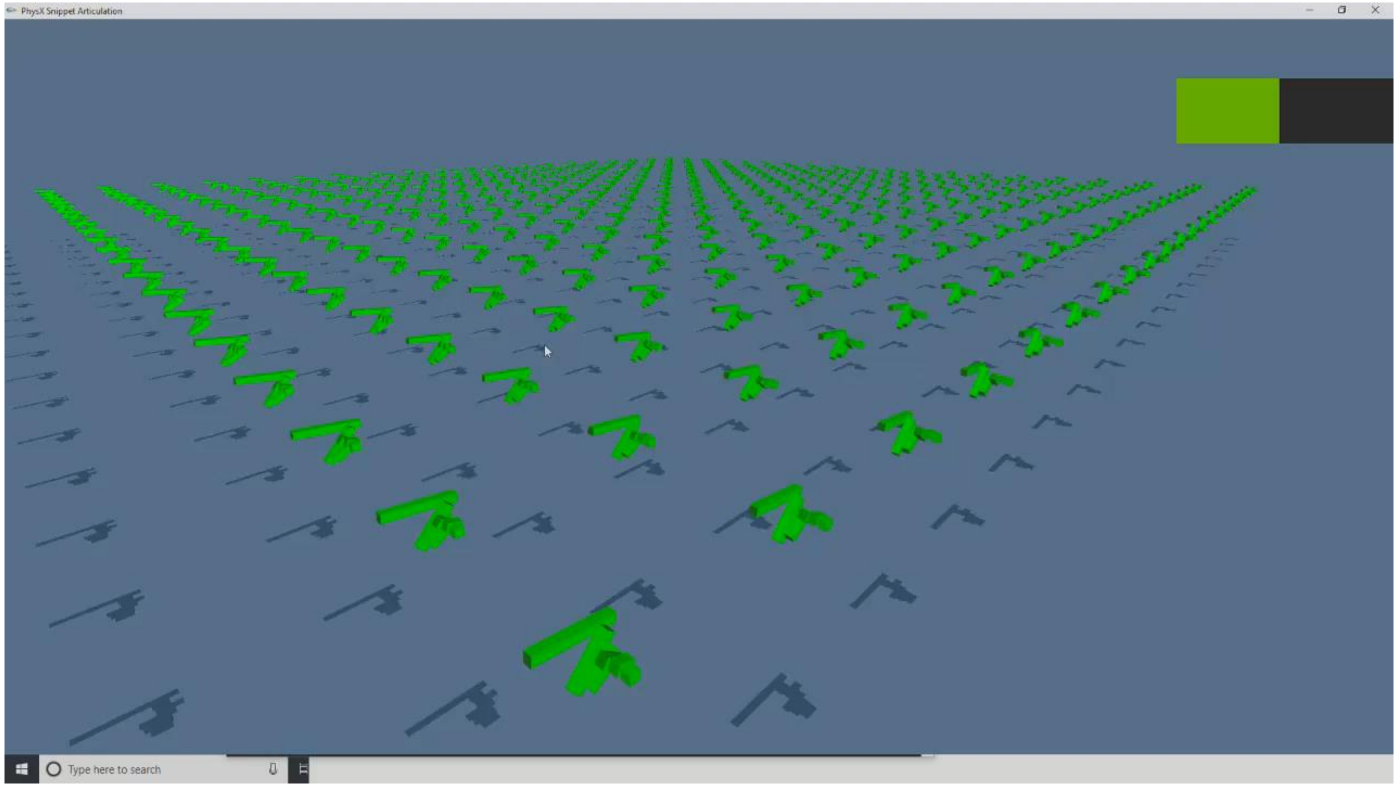
IMMEDIATE MODE PERFORMANCE

1 Articulation



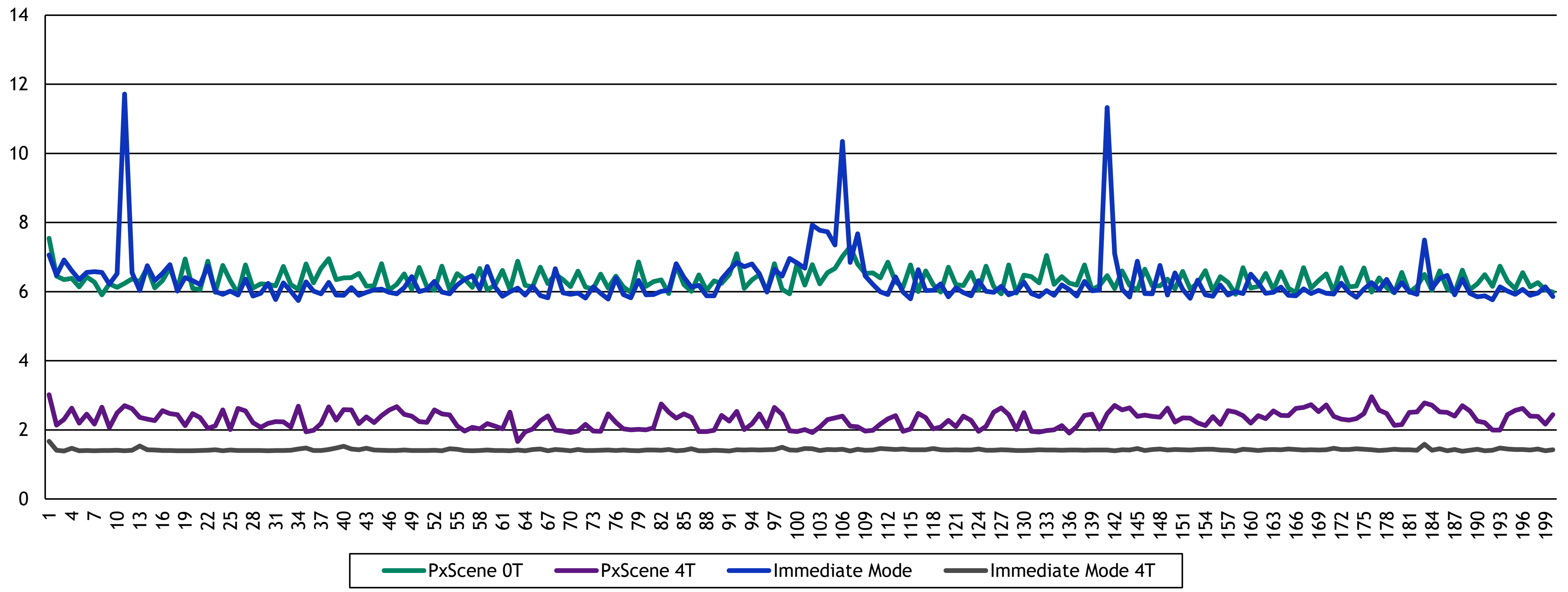
MULTIPLE INDEPENDENT SIMULATIONS

400 simple articulations



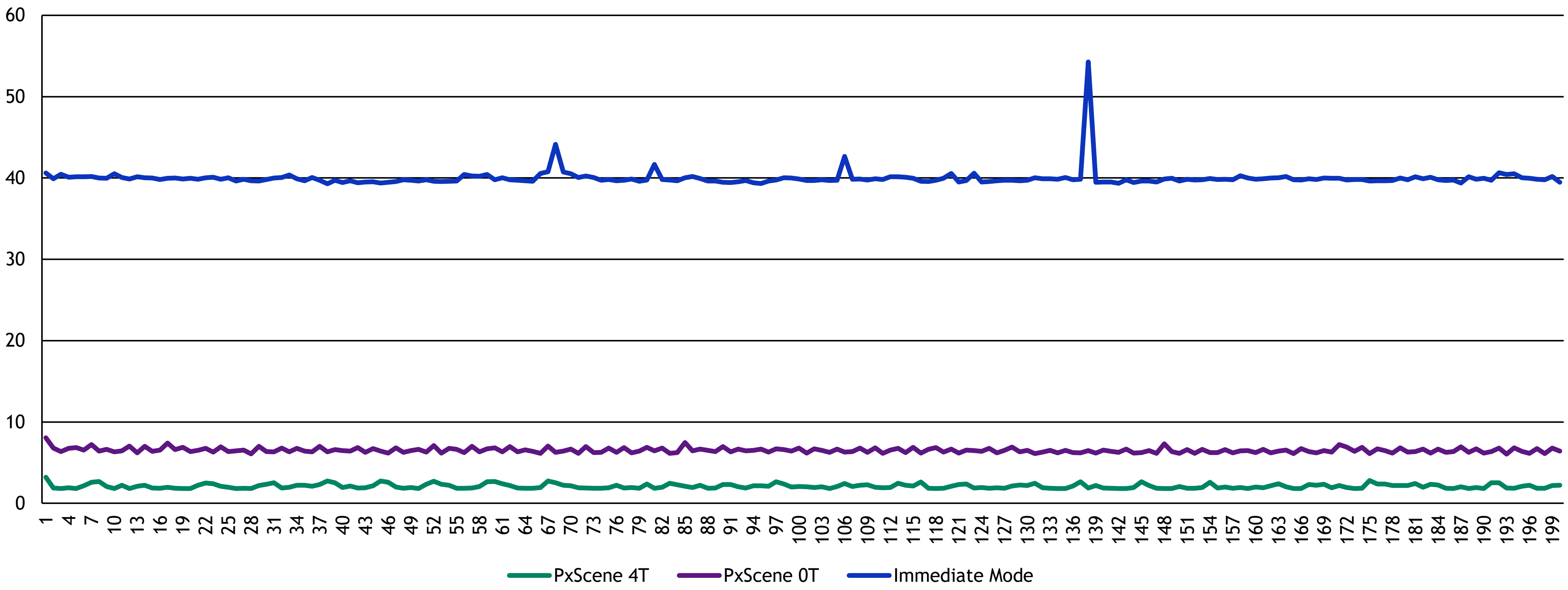
IMMEDIATE MODE PERFORMANCE

400 Articulations



WHEN EVERYTHING INTERACTS!

400 Articulations



GPU ARTICULATIONS

Work-in-progress

Extends GPU PGS and TGS Rigid Body solver to support articulations

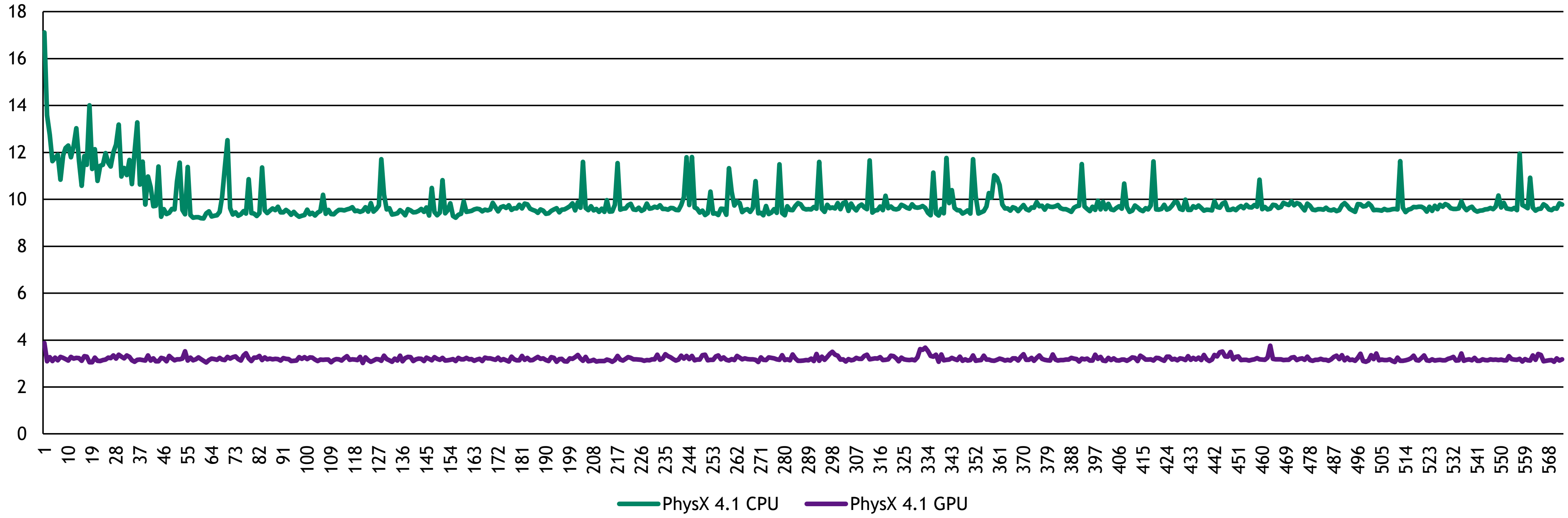
Supports mixed simulation (rigid bodies + articulations)

Provides significant performance uplift compared to CPU-based articulations

Ideally-suited to simulations of 100s or 1000s of articulations and rigid bodies

GPU ARTICULATION PERFORMANCE

1800 swinging 18-dof chains with limits



GPU ARTICULATIONS

Future work

Faithful port of PhysX 4.0 articulation solver completed

Multi-body interaction between articulations and rigid bodies/other articulations complete

GPU version of optimized PhysX 4.1 contact/joint solver currently in the works

Articulation support in GPU TGS solver in progress

PHYSX FOR GAMES

PxScene simulation provides a scalable, asynchronous n-body simulation

Ideally-suited for simulating the main gameplay-effecting bodies

Immediate mode ideally suited for smaller-scale physics simulation

Local secondary physics for characters, network replays etc.

GPU simulation can be used to scale simulations massively

e.g. scalable destruction, server-side destruction/simulation etc.

NON-GAMING APPLICATIONS

Industrial simulation

Robotics

Reinforcement learning

- Immediate mode for low-latency small-scale simulations
- GPU articulations for larger-scale simulations

Model-based control

- Forward and inverse dynamics

Visual FX

BEYOND GAME VEHICLE PHYSICS?

Gordon Yeoman, Developer | 14 March, 2019 | Booth S466| South Hall

OUTLINE

PhysX Vehicles are now being used in non-gaming vehicle simulation applications such as NVIDIA DriveSim.

Using games technology in non-gaming applications leads to an obvious question.

> Q: What is the integrity of the PhysX vehicle model in scenarios currently of interest?
>
> A: Let’s compare with world class, industry-standard vehicle dynamics simulators and analyse the results.

PHYSX VEHICLE FEATURE SET

What is a PhysX vehicle?

- Engine (1-d rigid body, drive torque curve, damping)
- Drivetrain (clutch, gearbox, final drive, differential)
- Suspension (spring-damper)
- Wheels (1-d rigid body)
- Tires (longitudinal/lateral slip angles/forces, non-linear response, friction)
- 3d rigid body (jointed to sidecars, trailers etc)

SIDE BY SIDE COMPARISION I

Test methodology

- Run PhysX side by side with leading 3rd party simulator used widely by automotive industry
- Configure vehicle data to be as closely matched as possible
- Run PhysX and leading 3rd party simulator and compare simulation results
- Test acceleration, braking, steering conditions that represent usual and safe driving behaviors

SIDE BY SIDE COMPARISION II

Test methodology

- Drive PhysX and 3rd party simulator with identical driving input (steer, brake, throttle) updated at 75Hz
- PhysX and 3rd party simulator drive on the same road geometry and friction
- Run PhysX and 3rd party simulator at 1KHz
- Sample simulation state (wheel rolling speed, vehicle speed) at 60Hz.
- Plot trajectory of maneuver

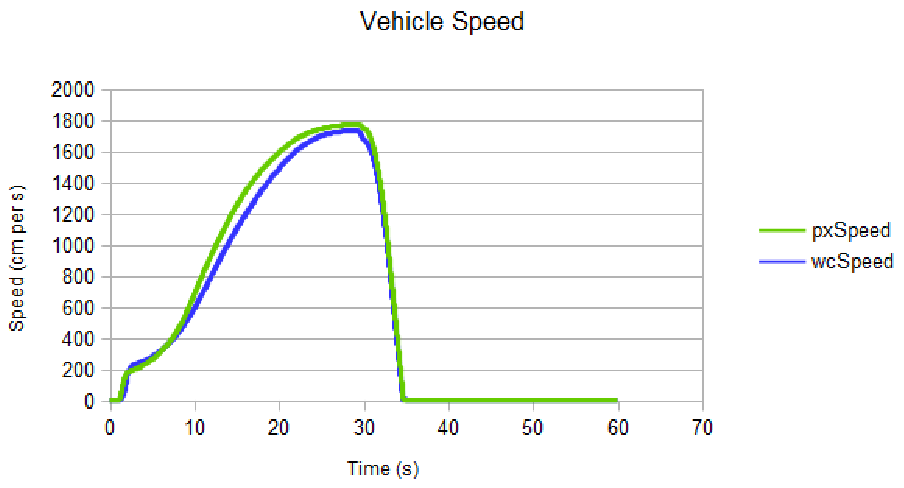
THROTTLE AND BRAKE

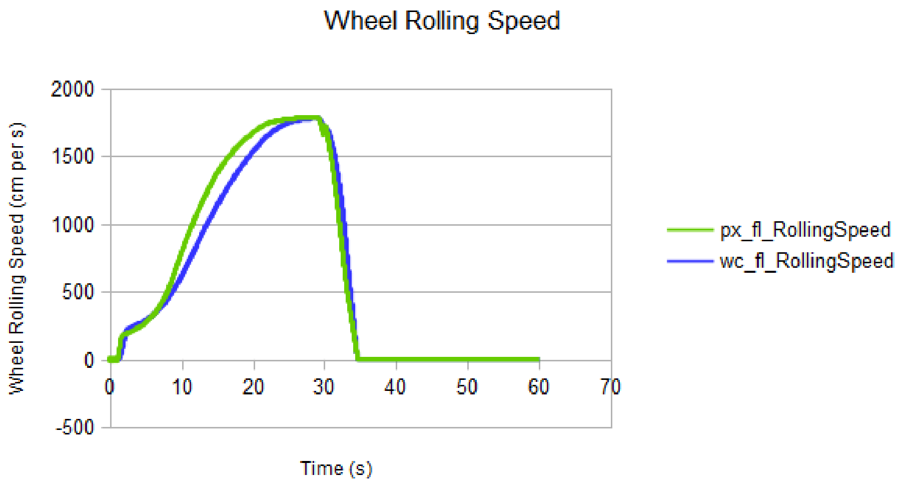
Rev the engine

Engage the clutch

Accelerate to top speed in 1st gear

Apply the brake at 30 seconds





LANE CHANGE MANEUVER

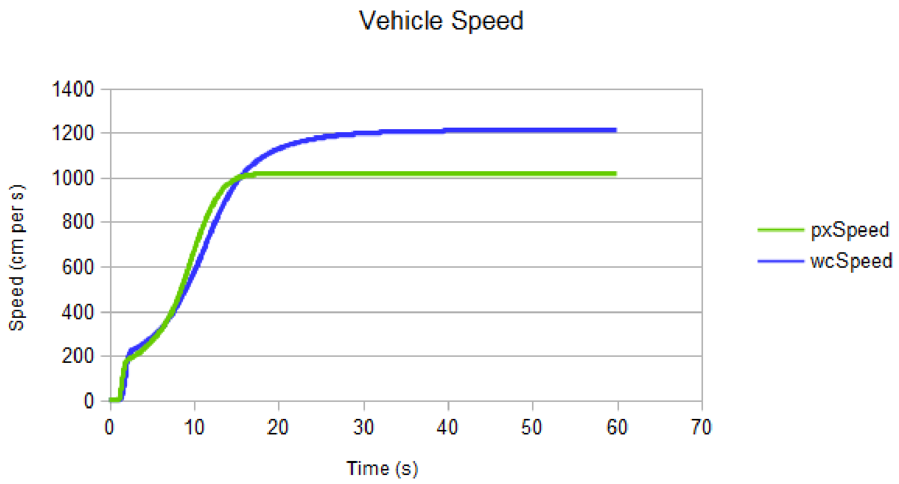
Rev the engine

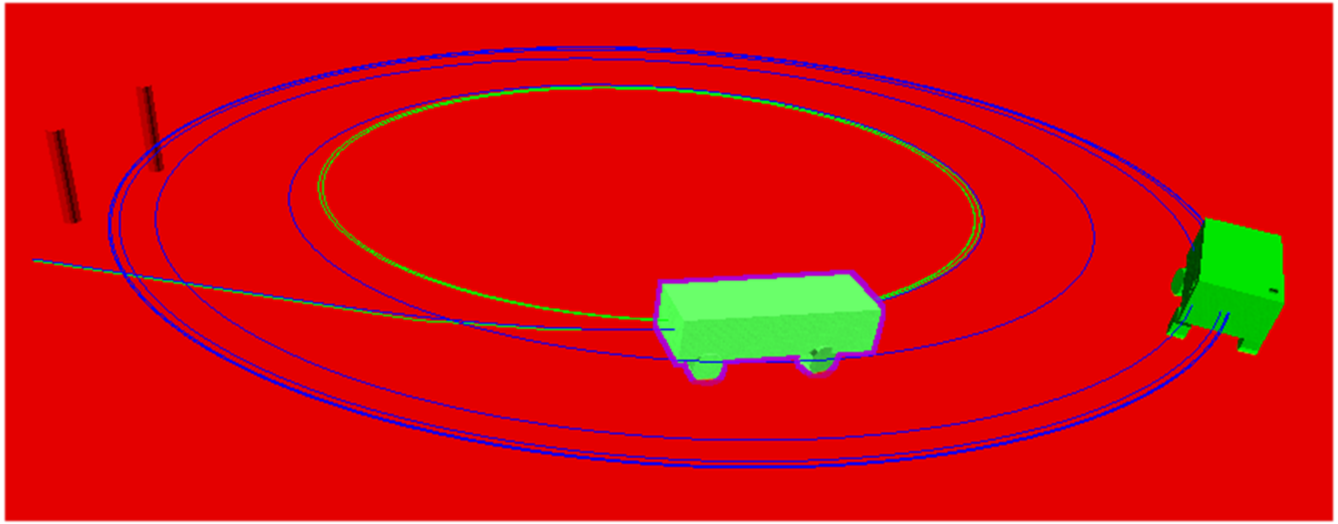
Engage the clutch

Throttle forwards

Steer maneuver to change lanes

(note: placeholder images)





QUESTIONS

Gordon Yeoman| gyeoman@nvidia.com

BEYOND GAME VEHICLE PHYSICS?

Gordon Yeoman, Developer | 14 March, 2019 | Booth S466| South Hall

OUTLINE

PhysX Vehicles are now being used in non-gaming vehicle simulation applications such as NVIDIA DriveSim.

Using games technology in non-gaming applications leads to an obvious question:

Q: What is the integrity of the PhysX vehicle model in scenarios currently of interest?

A: Let’s compare with world class, industry-standard vehicle dynamics simulators and analyse the results.

PHYSX VEHICLE FEATURE SET

What is a PhysX vehicle?

- Engine (1-d rigid body, drive torque curve, damping)
- Drivetrain (clutch, gearbox, final drive, differential)
- Suspension (spring-damper)
- Wheels (1-d rigid body)
- Tires (longitudinal/lateral slip angles/forces, non-linear response, friction)
- 3d rigid body (jointed to sidecars, trailers etc)

SIDE BY SIDE COMPARISION I

Test methodology

- Run PhysX side by side with leading 3rd party simulator used widely by automotive industry
- Configure vehicle data to be as closely matched as possible
- Run PhysX and leading 3rd party simulator and compare simulation results
- Test acceleration, braking, steering conditions that represent usual and safe driving behaviors (accelerate, brake, lane change steering)

SIDE BY SIDE COMPARISION II

Test methodology

- Drive PhysX and 3rd party simulator with identical driving input (steer, brake, throttle) updated at 75Hz
- PhysX and 3rd party simulator drive on the same road geometry and friction
- Run PhysX and 3rd party simulator at 1KHz
- Sample simulation state (wheel rolling speed, vehicle speed) at 60Hz.
- Plot trajectory of maneuver

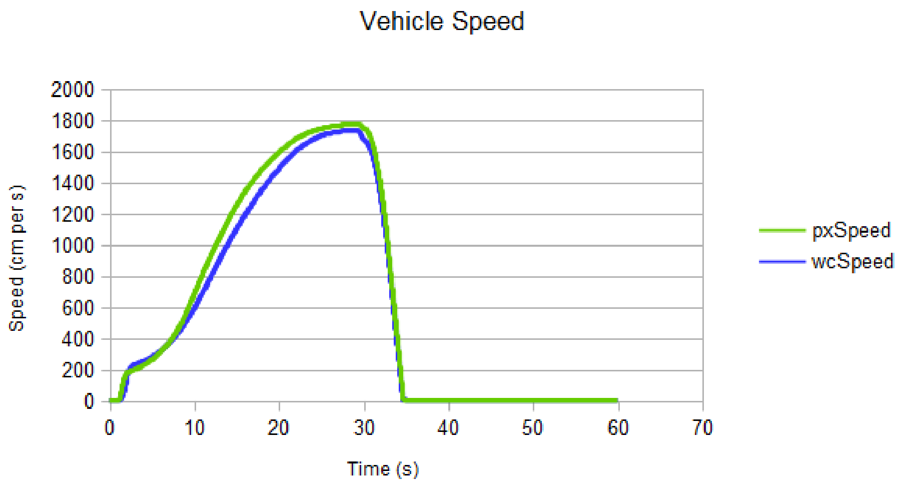
THROTTLE AND BRAKE

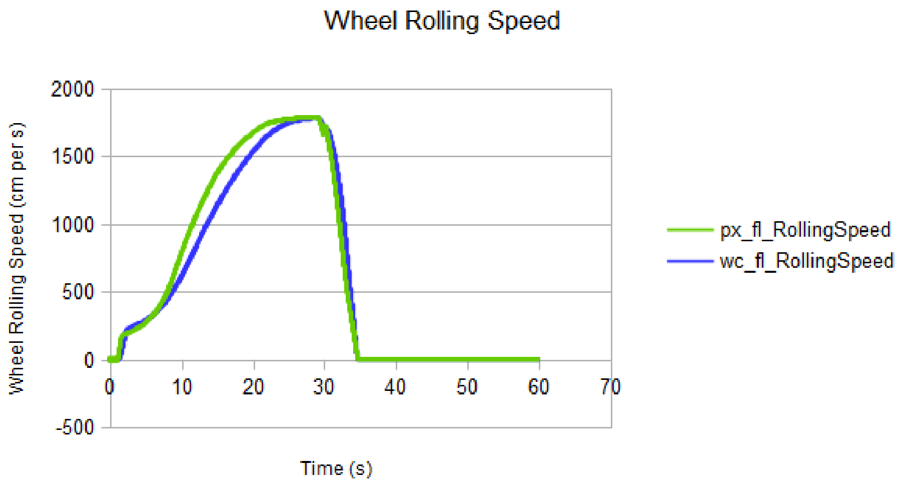
Rev the engine

Engage the clutch

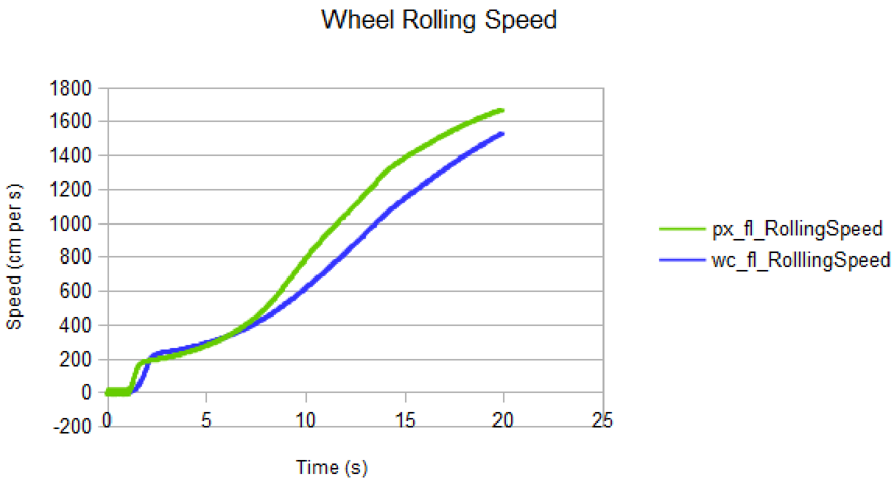
Accelerate to top speed in 1st gear

Apply the brake at 30 seconds





LANE CHANGE MANEUVER



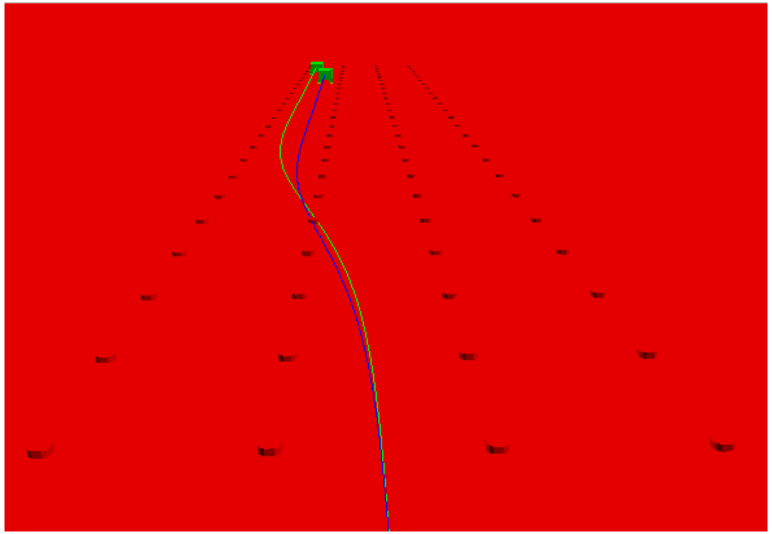
Rev the engine

Engage the clutch

Throttle forwards

Lane change maneuver begins at 10s

Lane change maneuver ends at 16s



QUESTIONS

Gordon Yeoman| gyeoman@nvidia.com