Course Work 2

Nile University

Scientific Computing

Ahmed Samr (1910032)

Youssef Ansara (1910063)

Amr Al Sayed (202000199)

### General Form

For explicit methods:

For implicit methods:

Coefficients:

### Forward Euler (1st Order)

### Explicit Midpoint (2nd Order)

### Heun (2nd Order)

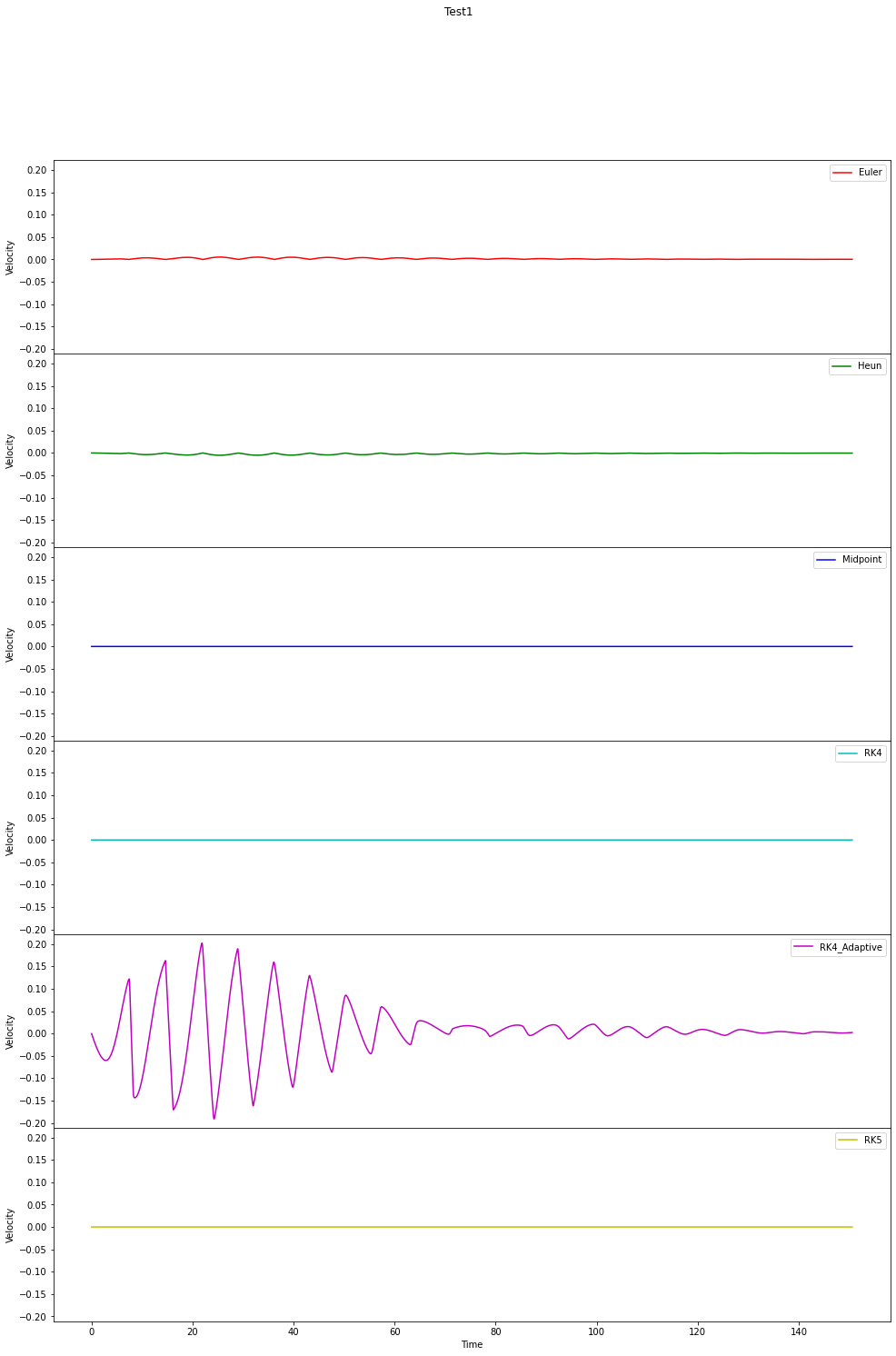
### Fourth-Order Runge–Kutta (4th order)

### Fifth-Order Butcher’s Runge-Kutta (6th order)

### Comparison of Integration methods

All of the integration methods are compared with these parameters

### Test 1

Figure Comparison of Average particle positions using Test 1

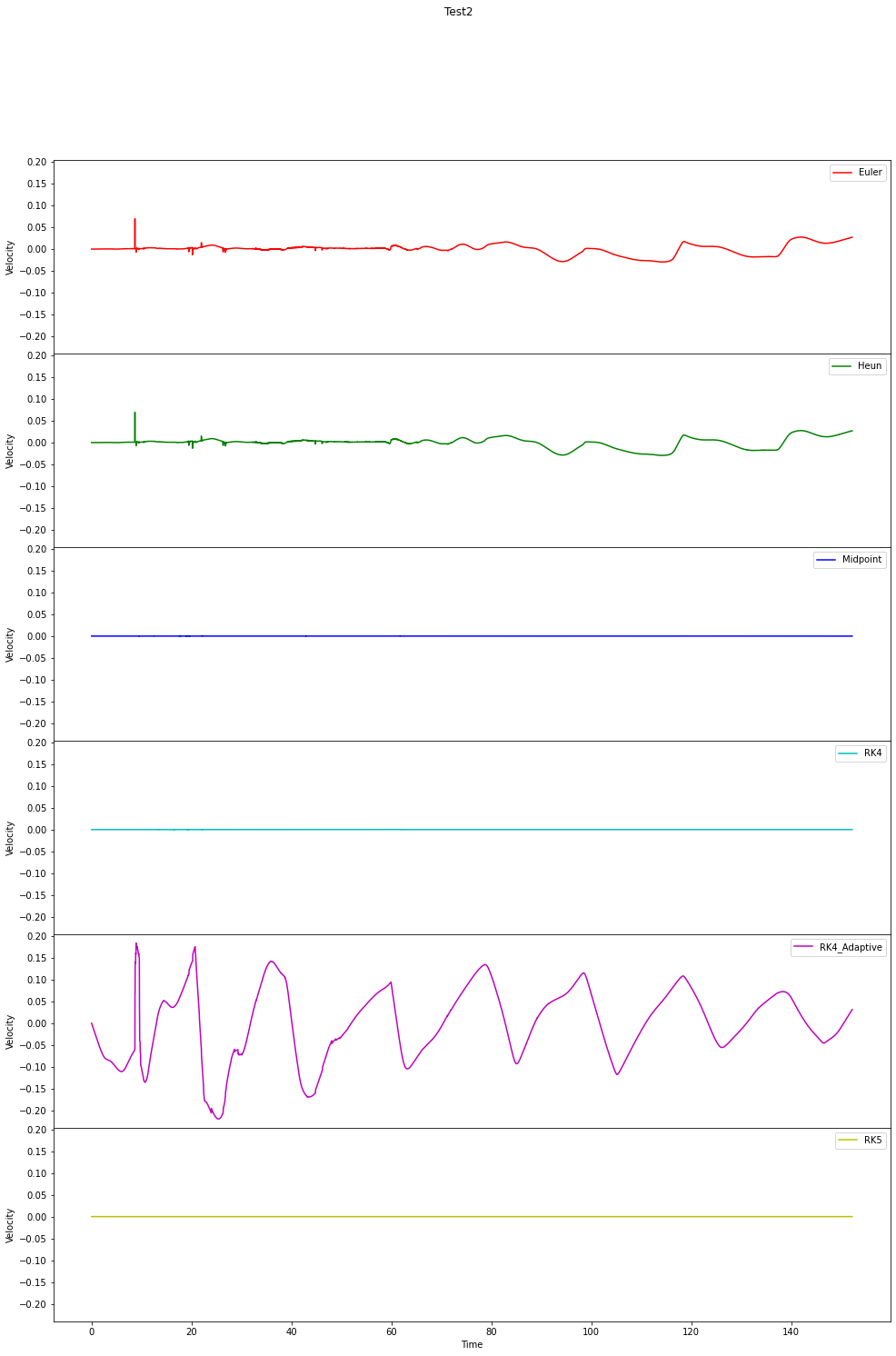
Test 2 

Figure Comparison of Average particle positions using Test 2

### Test 3

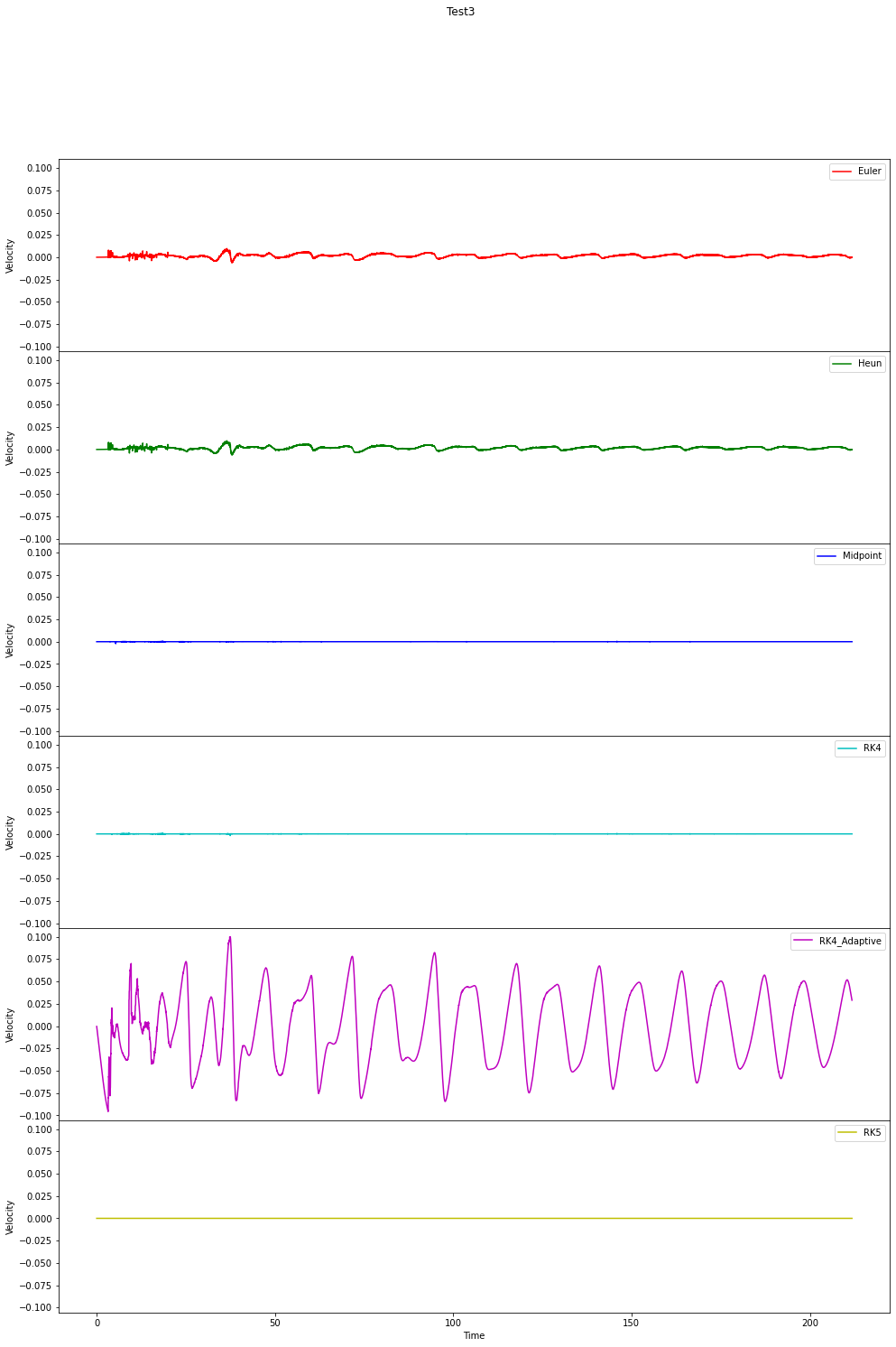


Figure Comparison of Average particle positions using Test 3

All comparisons were made with Fifth Runge-Kutta as the base. The average position and velocity of all the particles in the test cloth was outputted with respect to time and the difference between the position/velocity of the model to be compared and Fifth Order Runge-Kutta is graphed. The position difference and the velocity difference graphs were identical so only velocity graphs are provided.

So, using the default parameters, all integrators behaved very similarly except for Adaptive Fourth Order Runge-Kutta method which fluctuates wildly away from the result of Fifth Order Runge-Kutta. Euler & Midpoint methods also show some deviation away from the Fifth Order Runge-Kutta result across all tests.

This means that for our next test we will **NOT** be decreasing the step size as the simulation is already very similar and a smaller test size will not show any differences. Also only Test 3 will be used as it is the most intensive one and has many collisions and external factors that strain the system.

Let’s take these parameters. They’re the default parameters but with 10 times the step size

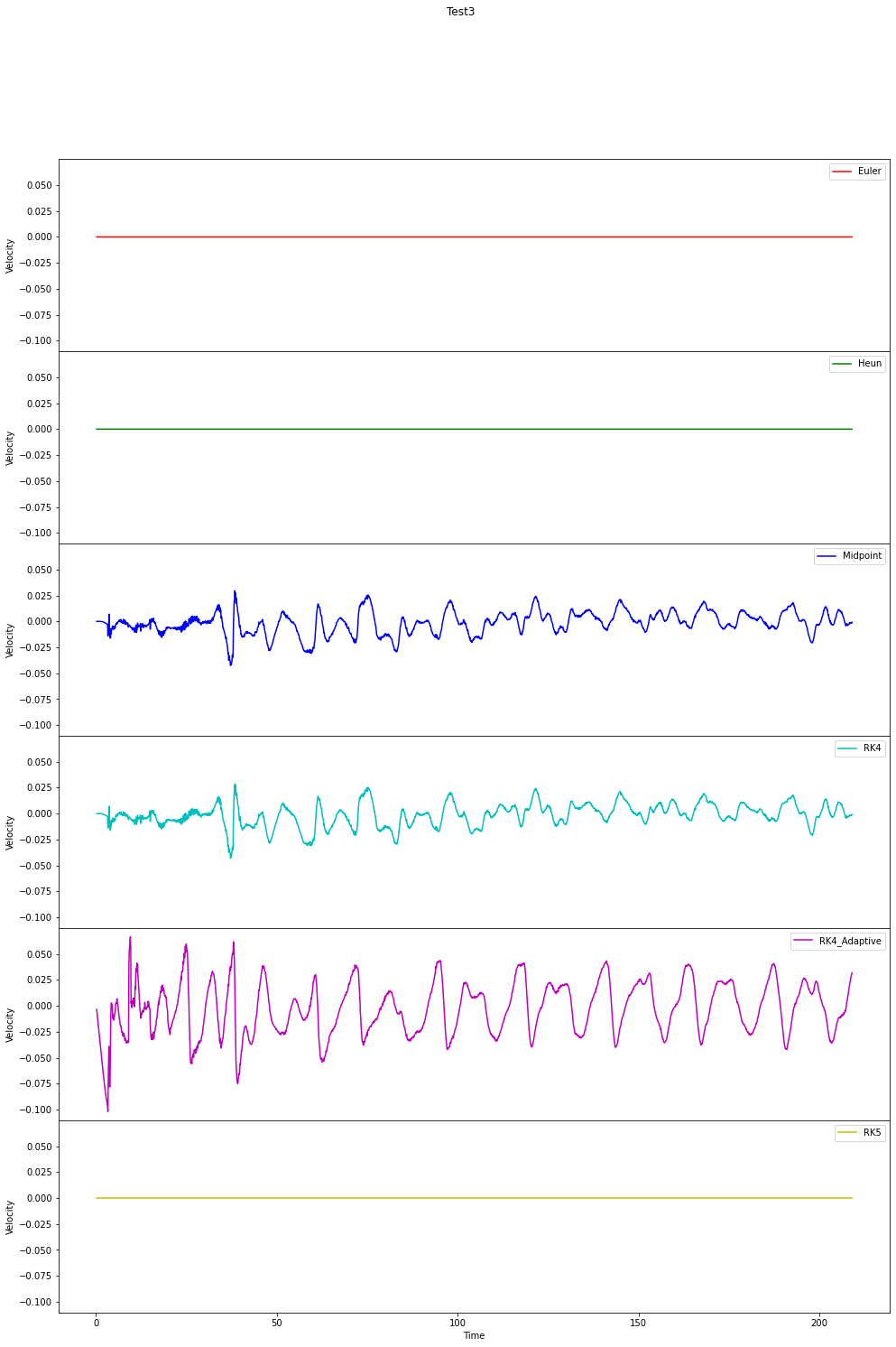


Figure Comparison of Integration methods with an increased step size

This result is very strange and is extremely unexpected. Euler matching almost exactly the Fifth Order Runge Kutta is not a sensible result. I failed to see the reason for this.

To test the limits of the system, we will be quadrupling the default Spring Constant to 20 which simulates very stiff springs between vertices.

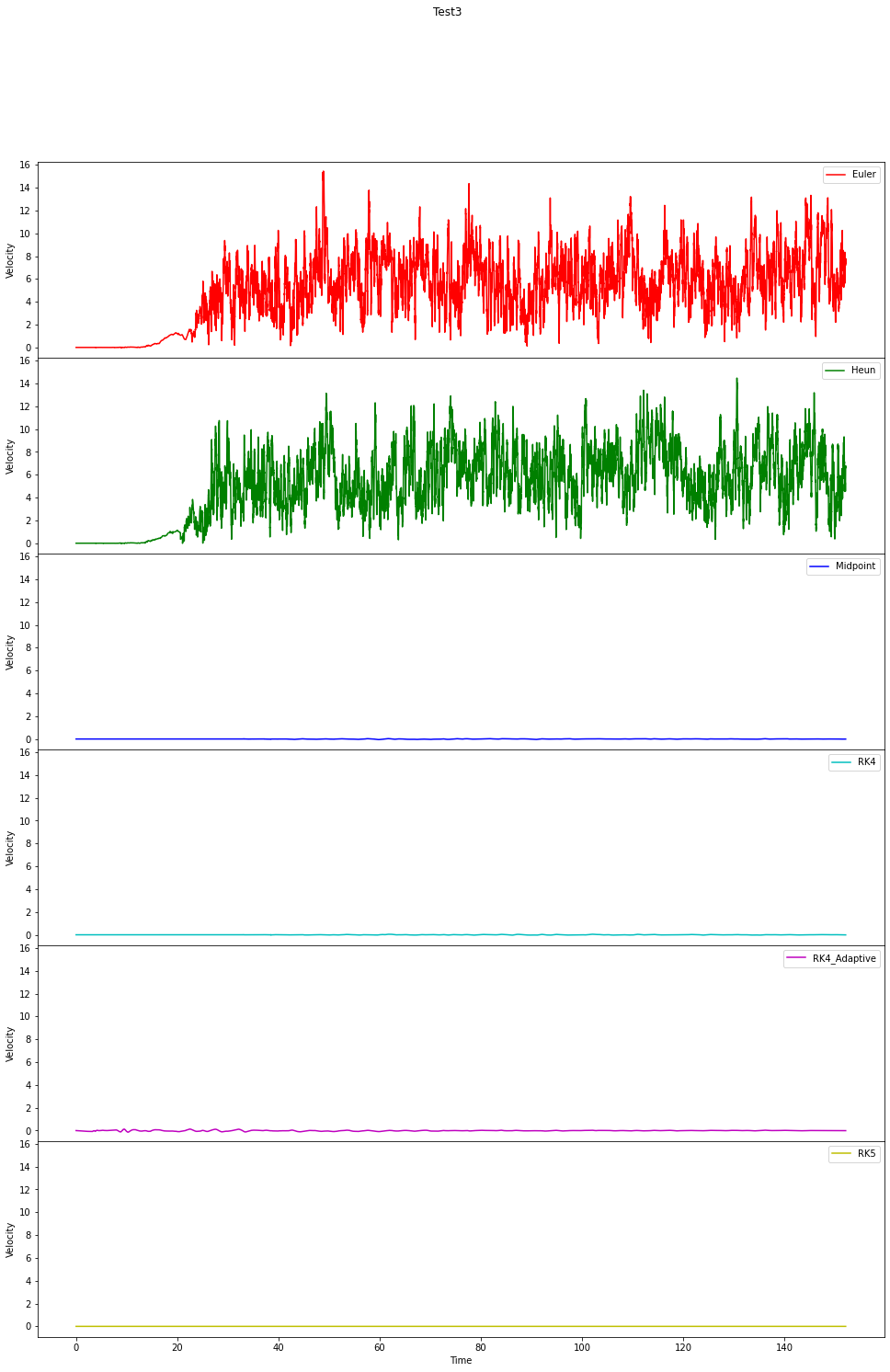


Figure Comparison of Integration method when the Spring Constant is 20

Both Euler & Huen start out as very similar to Fifth Order Runge-Kutta but as the collisions increase the system just shakes vigorously in a way that is nothing like a cloth. We conclude from this that primitive integration methods like Euler & Huen are not suitable if the cloth that is to be simulated is very rigid. A good compromise would be to use Midpoint method as it performs reasonably well across all test and changing factors but it is only a 2nd order Runge-Kutta method so it doesn’t require a lot of computations.