

# Proposed Feature

*Group 4*

---

## Abstract

In an xml input file, the user will specify species involved in the reaction, the chemical equations, and stoichiometric coefficients, the rate coefficient parameters, and whether the reaction is reversible. Because chemical kinetics involves a series of differential equations, our team will be implementing differential equations (ODE) solvers. Using a class/method(which do we want to do?), the user can specify a scipy solver and desired visualization.

Users will be able to choose their integrator. The scipy.integrate library has two powerful routines, ode and odeint, for numerically solving systems of coupled first order ordinary differential equations (ODEs). Odeint (ODE integrator) is robust and can handle both stiff and non-stiff problems. If users chose ODE, we will allow the user to choose from the following solvers, depending on the power of their equation:

- RK45 (Explicit Runge-Kutta method of order 5(4))
- RK23 (Explicit Runge-Kutta method of order 3(2))
- Radau (Implicit Runge-Kutta method of Radau IIA family of order 5.)
- BDF (Implicit method based on Backward Differentiation Formulas.)
- LSODA (Adams/BDF method with automatic stiffness detection and switching)

As we develop the code, we will test these to ensure they provide the appropriate output. Once the solver has been implemented, solutions may be stored as XML, HDF5, or ASCII.

We will then write a visualization library for the output of ODE (in SQL, yes? Or do we want to have the ODE print to xml?) using matplotlib, pandas and a social network analysis package called NetworkX. Users will be able to plot concentrations by time at a fixed temperature, and chose the

number of species per plot. Users may choose time axis (number of ticks). The output will be a HDF5 file.

We will also include a social network analysis visualization, using NetworkX, to demonstrate what species are reacting with one another (i.e. any reaction that has O<sub>2</sub> in it has a branch to the products/reactants that use O<sub>2</sub>). These will be implemented with a color scale to demonstrate products and reactants. The output will be a HDF5 file. All external dependencies include: matplotlib, pandas, NetworkX and scipy.integrate.

---