

# Hierarchical approach to ODE solving

*increasing sophistication* →

↑  
*developed from the ground up*

↓  
*executes from the top down*

**Concept**

**CCK example**

**Numerical Recipes in C**

**Octave**

Your main program (initial conditions, etc.)

Nbody\_test2

int main{ }

General purpose front-end

adaptive\_stepper

odeint

lsode, lsode\_options

Integrate from t1 to t2, using as many steps as needed

adaptive

rkqs

Double step with increase of order and error estimate

twostep (generic)

bsstep,  
stiff,  
stifbs

adams,  
BDF

single, fixed step forward in time

rk2, euler

rkck

Equations of motion

harmonic, goofy, Nbody

- It is common practice to develop a library of solvers rather than a single algorithm for solving ODEs.
- These solvers have many features in common, across multiple developers and languages.
- LSODE (Livermore Solver for Ordinary Differential Equations)
  - Widely used and published.
  - Part of ODEPACK, written in FORTRAN.
- Numerical Recipes code makes it difficult to get solution at desired times.
  - Advice: Write your own (simple) front end for their front end.
- MATLAB has a variety of solvers: ode45, ode15i, ode113, ode23, ode23s, ode23t, ode23b.