Review of First Computer Labs

- For the predator-prey model, we *should* expect to see closed loops of the solutions.
  - ◇ Using fixed-step methods, such as *myeuler*, we see, instead, spirals.
  - ◇ Smaller step sizes can only delay the happening of spiral behavior, but cannot prevent it. Regardless the sizes of the steps taken and regardless the order of the methods, fixed-step methods cannot fix the problem.
  - ◇ Using higher order variable-step methods, such as em ode45, the trajectories of solutions appear to be closed.
- Higher order and variable step methods are the way to solve ordinary differential equations numerically.
  - ◇ These methods use smaller step sizes at difficult spots. This will improve the accuracy.
  - ◇ These methods use larger step sizes at easy regions. This will increase the speed.
  - ◇ The key to the design of variable step method is a *algorithmic brain* that can determine whether a difficult or a easy region is being integrated over. We need an error estimator.
- The MATLAB routines *ode23* and *ode45* are using variable step Runge-Kutta methods as the integrator. More precisely, *ode45* is using a fourth order RK method to do the integration while using a fifth order RK method to estimate the error. (You don't need to worry about the details in this course.)
- There are many other types of variable step methods. See my WEB page in the directory */Teaching/Courses/F97/MA302/Library*.

Moving toward Runge-Kutta Formulas

- From the Fundamental Theorem of Calculus, we know

$$y(x+h) = y(x) + \int_x^{x+h} f(t, y(t))dt.$$

  - ◇ If we can improve the approximation of $f(t, y(t))$, then we can improve the integrator.
  - ◇ Midpoint rule:

$$\begin{aligned} f(t, y(t)) &\approx f\left(x + \frac{h}{2}, y(x + \frac{h}{2})\right) \\ y_{n+1} &= y_n + hf\left(x_n + \frac{h}{2}, y(x_n + \frac{h}{2})\right) \\ y(x_n + \frac{h}{2}) &\approx y(x_n) + \frac{h}{2} f(x_n, y(x_n)) \\ y_{n+1} &= y_n + hf\left(x_n + \frac{h}{2}, \mathbf{y_n} + \frac{\mathbf{h}}{\mathbf{2}}\mathbf{f}(\mathbf{x_n}, \mathbf{y_n})\right) \end{aligned}$$

    - ▷ Note that there are two function evaluations involved.
    - ▷ An Euler step is used to estimate the solution at the midpoint, at which the slope is used again as an Euler step to re-shoot from (x_n, y_n).
    - ▷ This is a 2-stage method.

Runge-Kutta Formulas

- A general R-stage Runge-Kutta method is defined by the one-step method:

$$y_{n+1} = y_n + h\phi(x_n, y_n, h) \tag{0.1}$$

where

$$\begin{aligned} \phi(x_n, y_n, h) &= \sum_{r=1}^{R} c_r k_r & (0.2)\\ \sum_{r=1}^{R} c_r &= 1 \\ k_r &= f(x_n + a_r h, y_n + h\sum_{s=1}^{R} b_{rs} k_s) & (0.3)\\ \sum_{s=1}^{R} b_{rs} &= a_r. \end{aligned}$$

- It is convenient to display the coefficients occurring in (0.2) and (0.3) in the following form, known as a *Butcher array*:

$$\begin{array}{c|cccc} a_1 & b_{11} & b_{12} & \dots & b_{1R} \\ a_2 & b_{21} & b_{22} & \dots & b_{2R} \\ \vdots & \vdots & & & \vdots \\ a_R & b_{R1} & b_{R2} & \dots & b_{RR} \\ \hline & c_1 & c_2 & \dots & c_R \end{array}$$

- Lots of research has been done in the past few decades to find out the *best* combinations of these coefficients. (If you like to know more about these, join the graduate school and be a scholar!)