We restrict attention to first-order systems of ODEs and pay special attention to linear and constant-coefficient systems:

An IVP for an ODE usually has the form \( y' = f(t, y) \) with initial value \((t_0, y(t_0))\). A linear ODE \( f \) has the form \( f(t, y) = A(t)y(t) \), and is said to have constant coefficients if \( A(t) \) does not vary with \( t \).

Existence and uniqueness of a solution to an IVP is guaranteed over any domain \( D \) on which \( f \) is Lipschitz continuous:

\[
\exists L \in \mathbb{R}, \| f(t, y) - f(t, \hat{y}) \| \leq L \| y - \hat{y} \|, \forall y, \hat{y} \in D
\]

which is stronger than continuity of \( f \) (differentiability of \( y \), but weaker than differentiability of \( f \), i.e. \( f \) can suddenly begin to change at a different rate. The constant \( L \) bounds the rate at which similar solutions \( y \) and \( \hat{y} \) can diverge or converge.
Convergence and Stability

- Generally, we seek to approximate $y(t_k)$ for a set of points $t_k = t_0 + kh$ by $\hat{y}_k$:
  
  Would like to converge with decreasing step-size, i.e. $\lim_{h \to 0} \hat{y}_k = y(t_k)$. Can measure global error (deviation from true solution) or local error (deviation of $\hat{y}_k$ from $u_{k-1}(t_k)$ where $u_{k-1}(t_{k-1}) = \hat{y}_{k-1}$ and $u'_{k-1} = y'$).

- Stability ascertains behavior as $t \to \infty$ either of the ODE itself or of a numerical method:
  
  - For an ODE, stability identifies convergence/divergence of perturbed solutions.
  - For a numerical method, it identifies step-size bounds necessary to ensure the method converges along with the ODE.
  - If a method is stable, the local error provides a trustworthy metric of global error.
  - A stiff ODE is a convergent ODE with rapidly varying small components, which poses a challenge for stability of methods.
Accuracy and Taylor Series Methods

- By taking a degree-$r$ Taylor expansion of the ODE in $t$, at each consecutive $(t_k, \hat{y}_k)$, we achieve $k$th order accuracy.

We can bound the local approximation error as the error the Taylor expansion,

$$
y(t_k + h) = y(t_k) + y'(t_k)h + \cdots + y^{(r)}(t_k)h^{r-1}/r!$$

which is $O(h^{r+1})$, which leads to $O(h^r)$ accuracy in the approximation to $f(t, y)$. Euler’s method is a first-order Taylor series method.

- Taylor series methods require high-order derivatives at each step:

  Analytic differentiation is expensive, so seek to approximate. Can perform a finite-differencing approximation by evaluating at points near $(t_k, \hat{y}_k)$ (multi-stage methods) or simply using previous points, e.g. $(t_{k-1}, \hat{y}_{k-1})$ (multi-step methods).
Multi-Stage Methods

- Multi-stage methods construct $\hat{y}_{k+1}$ by approximating $y$ between $t_k$ and $t_{k+1}$:
  
  \[
  \hat{y}_{k+1} = \hat{y}_k + h \left[ f(t_k, y_k) / 2 + f\left(t_k + h, y_k + h f(t_k, y_k)\right) / 2 \right].
  \]

  We can think of the above method as employing the trapezoid quadrature rule. The difference between Heun’s method and the (implicit) trapezoid method is that we evaluate at $f(t_k + h, y_k + hv_1)$ rather than working with the implicit value of $f(t_k + h, y_{k+1})$.

- The 4th order Runge-Kutta scheme is particularly popular:
  
  This scheme uses Simpson’s rule

  \[
  \hat{y}_{k+1} = \hat{y}_k + (h/6)(v_1 + 2v_2 + 2v_3 + v_4)
  \]
  \[
  v_1 = f(t_k, y_k), \quad v_2 = f(t_k + h/2, y_k + (h/2)v_1),
  \]
  \[
  v_3 = f(t_k + h/2, y_k + (h/2)v_2), \quad v_4 = f(t_k + h, y_k + hv_3)
  \]
Runge-Kutta Methods

Runge-Kutta methods evaluate $f$ at $t_k + c_i h$ for $c_0, \ldots, c_r \in [0, 1]$. A Runge-Kutta method can be derived as a quadrature rule

$$u_k(t_{k+1}) = \hat{y}_k + \int_{t_k}^{t_k+h} f(s, y(s))ds \approx \hat{y}_k + h \sum_{i=0}^{r-1} w_i f(t_k + c_i h, \hat{y}_{k,i})$$

where $\{(c_i, w_i)\}_{i=0}^{r}$ are quadrature node, weight pairs, but we still have flexibility in choosing $\hat{y}_{k,i}$. One good choice is to successively construct

$$\hat{y}_{k,i} = \hat{y}_k + h \sum_j a_{ij} f(t_k + c_i h, \hat{y}_{k,j}) = \hat{y}_k + h c_i f(t_k + c_i h, \hat{y}_{k,i-1})$$

More general choices for $a_{ij}$ are often represented by a Butcher tableau,

$$\begin{array}{c|cccc}
  c & A & 0 \\
  \hline
  w^T & 1/2 & 1/2 & 1/2 & 1/6 \\
  1/2 & 0 & 1/2 & 1/3 & 1/3 & 1/6 \\
  1/2 & 0 & 0 & 0 & 1 & 1/3 & 1/3 & 1/6 \\
  1/2 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/6 \\
\end{array}$$

e.g. for RK4,

If $a_{ij} = 0$ for $j \geq i$, the scheme is explicit, if for $j > i$ then diagonally implicit, and otherwise implicit.
Properties of Runge-Kutta and Extrapolation Methods

- Runge-Kutta methods are self-starting, but are harder to use to obtain error estimates. 

  Self-starting means that we only need $\hat{y}_k$ to form $\hat{y}_{k+1}$. Embedded Runge-Kutta schemes provides 4th + 5th order results, yielding an error estimate.

- Extrapolation methods achieve high accuracy by successively reducing step-size.

  Use single-step method with step sizes $h, h/2, h/4, ...$ to approximate solution at $t_k + h$. 
Multistep Methods

- Multistep methods employ \( \{\hat{y}_k\}_{i=0}^{k} \) to compute \( \hat{y}_{k+1} \):

  \[
  \hat{y}_{k+1} = \sum_{i=1}^{m} \alpha_i \hat{y}_{k+1-i} + h \sum_{i=0}^{m} \beta_i f(t_{k+1-i}, y_{k+1-i}).
  \]

  Interpolation is used to determine each \( \alpha_i \) and \( \beta_i \), method is explicit if \( \beta_0 = 0 \).

- Multistep methods are not self-starting

  However, they require few function evaluations. Multivalue methods generalize multistep methods to non-uniformly-spaced points.