ES440/ES911: CFD

Chapter 6. Methods for Unsteady Problems

Dr Yongmann M. Chung
`http://www.eng.warwick.ac.uk/staff/ymc/ES440.html`
`Y.M.Chung@warwick.ac.uk`

School of Engineering & Centre for Scientific Computing
University of Warwick

Chapter 6
Methods for Unsteady Problems

6.1 Introduction

- In computing unsteady flows, we have a fourth coordinate direction to consider: ***time***.
  - Time grid (FDM) & time volumes (FVM).
- The major difference between the space and time coordinates lies in ***the direction of influence***.
  - A force at any space location may influence the flow anywhere else.
  - Forcing at a given instant will affect the flow only in the future - no backward influence.
- All solution methods advance in time in a step-by=step or marching manner.

6.1 Introduction -Cont’d

- Unsteady flows are parabolic-like in time.
  - No conditions can be imposed on the solution except at the boundaries at any time after the initiation of the calculation.
- These methods are very similar ro ones applied to initial value problems for ODEs.

Ordinary Differential Equations, ODE

Ordinary Differential Equations, ODE

- Newton’s second law of motion

$$\boxed{\frac{dv}{dt} = \frac{F}{m}}$$

- Fourier’s heat law

$$\boxed{q = -k\frac{dT}{dx}}$$

- Fick’s law of diffusion

$$\boxed{J = -D\frac{dc}{dx}}$$

- For many equations, it is impossible to find analytic solutions and numerical methods are used for obtaining approximate solutions.

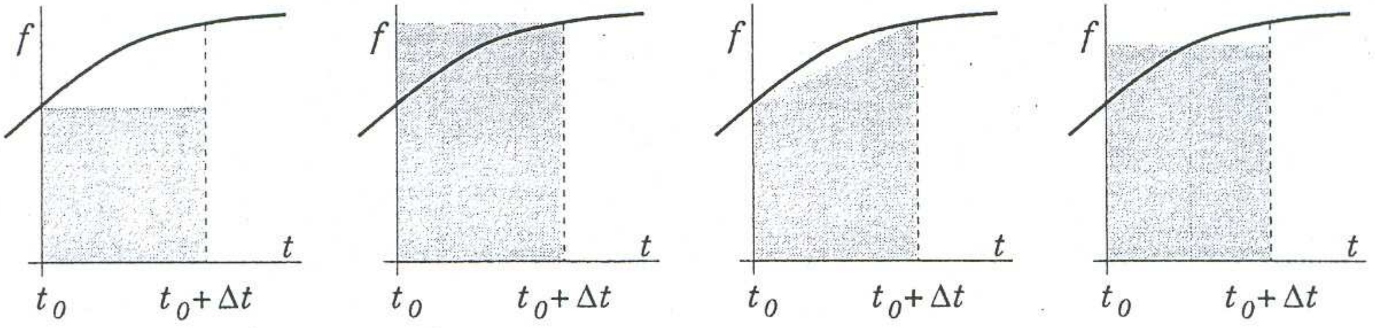
6.2.1 Tow-Level Methods

- Consider the first order ordinary differential equation with an initial condition:

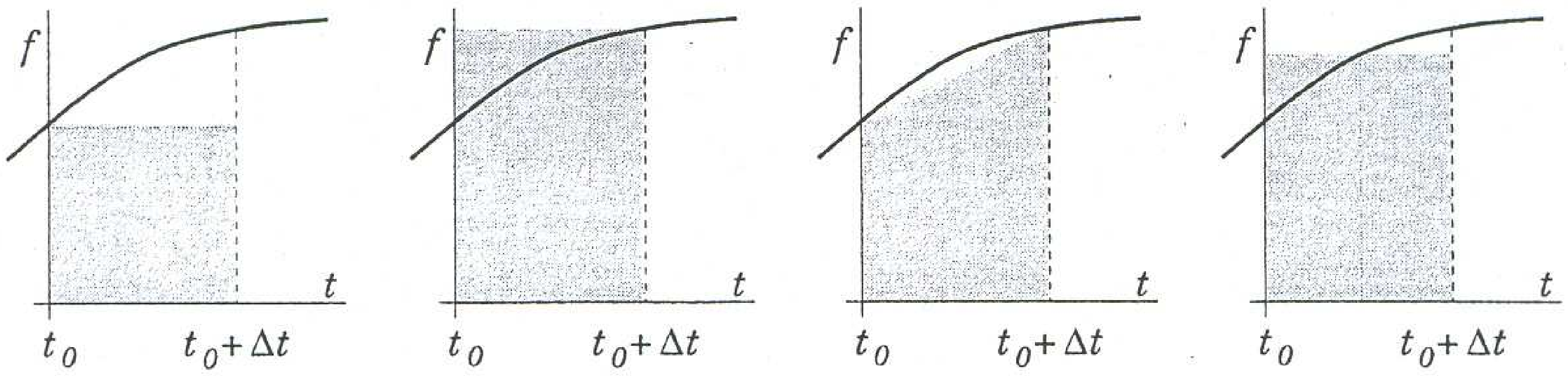
$$\frac{d\phi}{dt} = f(t, \phi(t)); \quad \phi(t_0) = \phi^0. \qquad (6.1^{F\&P})$$

- By integrating Eqn. (6.1) from t_n to $t_{n+1} = t_n + \Delta t$:

$$\int_{t_n}^{t_{n+1}} \frac{d\phi}{dt}\,\mathrm{d}t = \phi^{n+1} - \phi^n = \int_{t_n}^{t_{n+1}} f(t, \phi(t))\,\mathrm{d}t, \qquad (6.2^{F\&P})$$



6.2.1 Tow-Level Methods - Cont’d 2



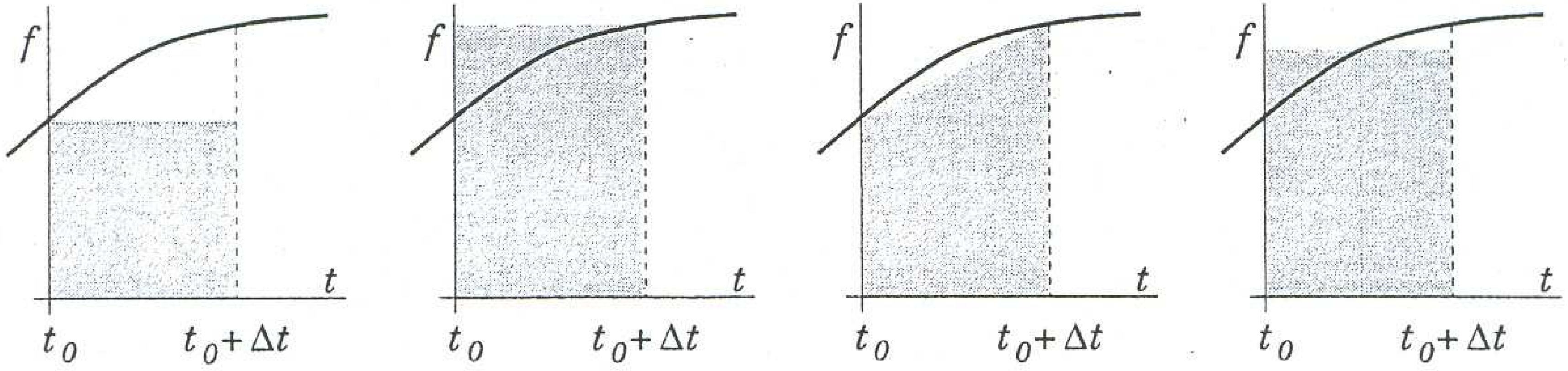
- The ***Explicit*** or ***Forward Euler*** Method

$$\phi^{n+1} = \phi^n + f(t_n, \phi^n)\Delta t. \qquad (6.3^{F\&P})$$

- The ***Implicit*** or ***backward Euler*** Method

$$\phi^{n+1} = \phi^n + f(t_{n+1}, \phi^{n+1})\Delta t. \qquad (6.4^{F\&P})$$

6.2.1 Tow-Level Methods - Cont'd 3



- The ***Crank-Nicolson*** Method using the ***trapezoid rule***

$$\phi^{n+1} = \phi^n + \frac{1}{2}\left[f(t_n, \phi^n) + f(t_{n+1}, \phi^{n+1})\right]\Delta t. \qquad (6.3^{F\&P})$$

- The ***midpoint rule*** Method - Basis of the ***Leapfrog*** Method.

$$\phi^{n+1} = \phi^n + f(t_{n+\frac{1}{2}}, \phi^{n+\frac{1}{2}})\Delta t. \qquad (6.4^{F\&P})$$

Euler Method

- Model 1st-order ODE equation.

$$\boxed{\frac{dy}{dx} = f(x,y), \;\; y(x_0) = y_0.}$$

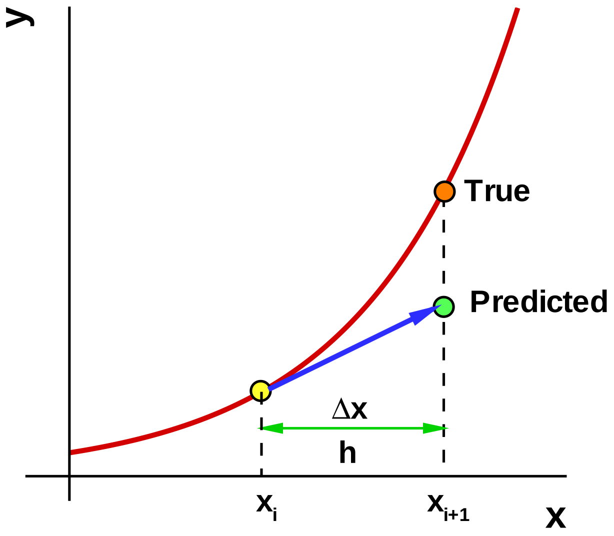
- Finite Difference Approximation.

$$\boxed{\frac{\Delta y}{\Delta x} = f(x,y).}$$

- Euler Method

$$\boxed{y_{i+1} = y_i + f(x_i, y_i)\Delta x_i,}$$

  - where $f(x_i, y_i)$ is the differential equation evaluated at x_i and y_i, and $\Delta x_i =$ the step size, or increment, in x.



$$\boxed{\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}.}$$

$$\boxed{\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}.}$$

- The error is $O(h)$: 1st Order.

Explicit Euler Method - Example

- For example, let's consider

$$\boxed{\frac{dy}{dx} = e^{2x}, \; y(0) = 1.}$$

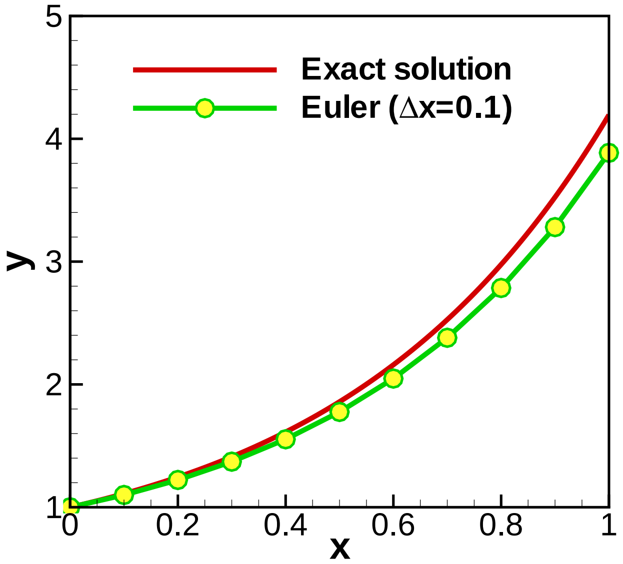
for $0 \leq x \leq 1$.

- The exact solution is

$$\boxed{y(x) = \frac{1}{2}e^{2x} + \frac{1}{2}.}$$

- Apply Euler Method

$$\boxed{y_{i+1} = y_i + e^{2x_i}\Delta x_i.}$$



Explicit Euler Method - Errors

- Truncation Error:

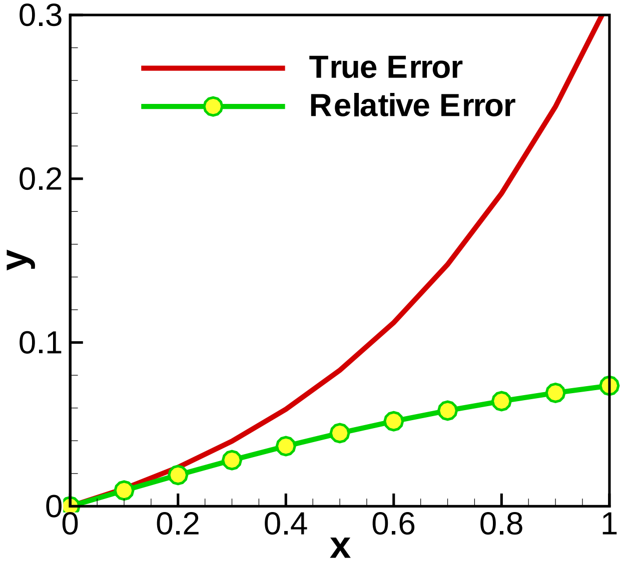
$$\boxed{\frac{dy}{dx} = \frac{\Delta y}{\Delta x}.}$$

- True Error

  = True Value - Approximation

- Relative Error

  =True Error / True Value



- The relative error is about 10%.
- Using small steps Δx.
- Using high-order methods.

Second-Order Runge-Kutta Methods

$$\boxed{\frac{dy}{dx} = f(x, y)}$$

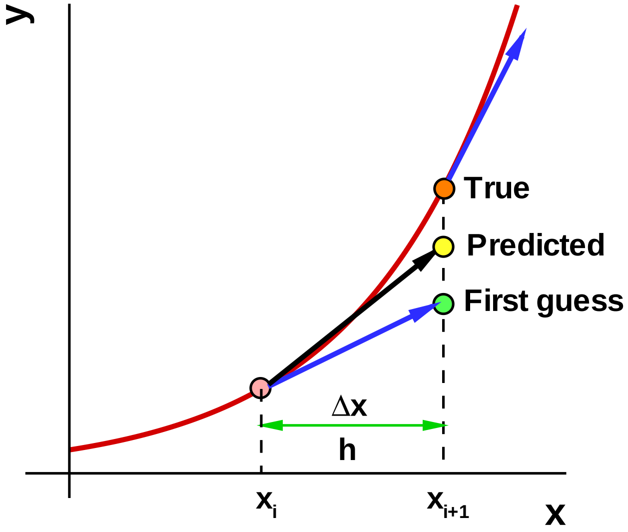
$$\boxed{y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)h}$$

where

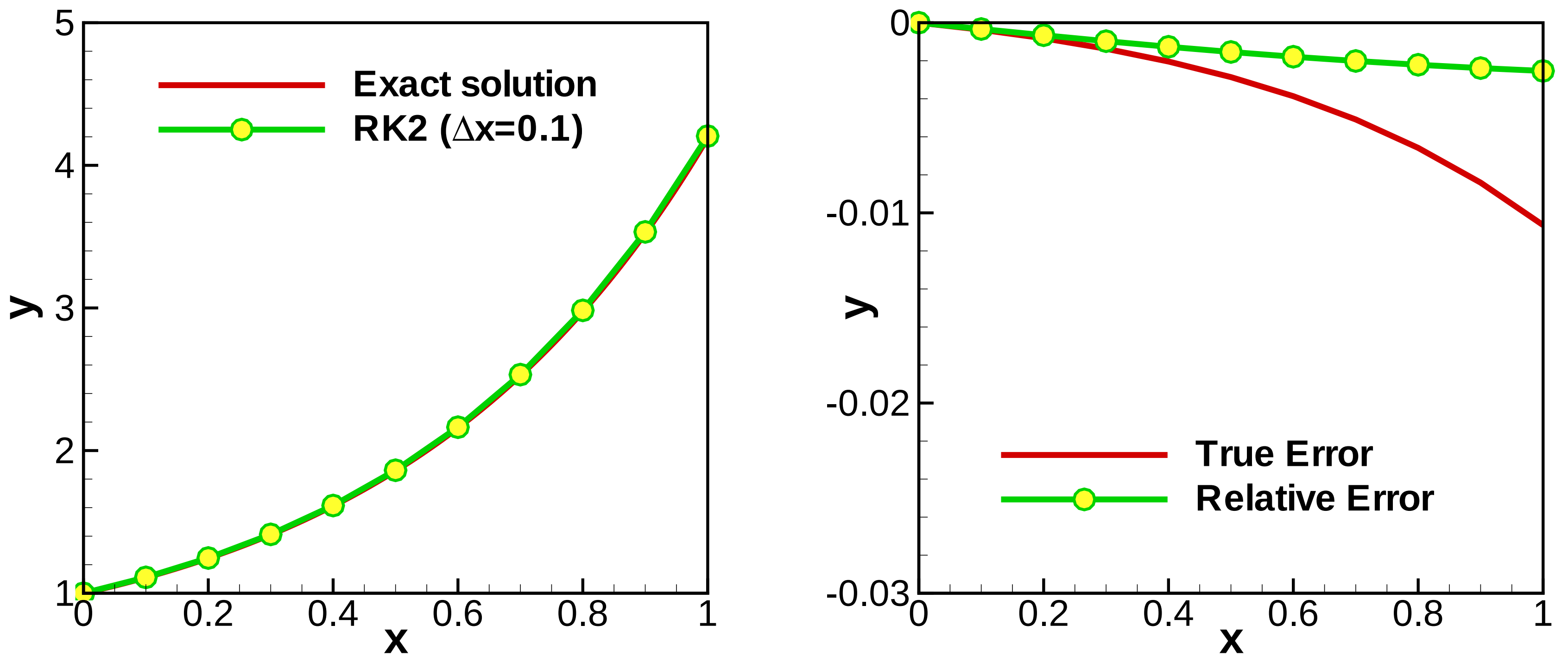
$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + h, y_i + k_1 h)$$

- The error is $O(h^2)$.



Second-Order Runge-Kutta Methods



- The relative error is less than 0.5%.
- Advantage of high-order methods

Third-Order Runge-Kutta Methods

$$\boxed{y_{i+1} = y_i + \frac{1}{6}(k_1 + 4k_2 + k_3)h}$$

where

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h) \\
k_3 &= f(x_i + h, y_i - k_1 h + 2k_2 h)
\end{aligned}$$

Fourth-Order Runge-Kutta Methods

$$\boxed{y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h}$$

where

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h) \\
k_3 &= f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h) \\
k_4 &= f(x_i + h, y_i + k_3 h)
\end{aligned}$$

Runge-Kutta Methods

- The major problem with Runge-Kutta methods
  - It is somewhat difficult to develop methods of very high order.
- An n-th order **R-K** method requires that the derivative be evaluated n times per time step,
  - making these methods more expensive than multipoint methods of comparable order.
- In partial compensation, the **R-K** methods of a given order are more accurate and more stable than the multipoint methods of the same order.
  - The coefficient of the error term is smaller.

ODE

- Model 1st-order ODE equation.

$$\boxed{\frac{dy}{dx} = f(x, y),\ y(x_0) = y_0.}$$

- Finite Difference Approximation.

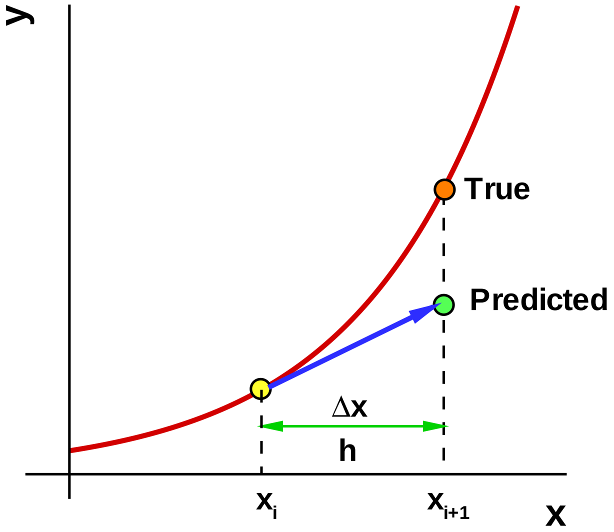
$$\boxed{\frac{\Delta y}{\Delta x} = f(x, y).}$$

- Euler Method

$$\boxed{y_{i+1} = y_i + f(x_i, y_i)\Delta x_i,}$$

  - where $\Delta x_i =$ the step size, or increment, in x.

- The error is $O(h)$: 1st Order.



$$\boxed{\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}.}$$

$$\boxed{\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}.}$$

Predator-Prey Models

- *Lotka-Volterra equations*

$$\boxed{\frac{dx}{dt} = ax - bxy}$$

$$\boxed{\frac{dy}{dt} = -cy + dxy}$$

  - $x =$ the number of prey
  - $y =$ the number of predators
  - $a =$ the prey growth rate
  - $c =$ the predator death rate
  - b and $d =$ the rate characterising the effect of the predator-prey interaction on prey death and predator growth, respectively.

Solution to *Lotka-Volterra equations*

$$\boxed{\frac{dx}{dt} = ax - bxy}$$

$$\boxed{\frac{dy}{dt} = -cy + dxy}$$

Use the following parameter values for the simulation:
$a = 1.2, b = 0.6, c = 0.8$, and $d = 0.3$.
Employ initial conditions of $x = 2$ and $y = 1$ and integrate from $t = 0$ to 30.

- Which methods?
- What time steps Δt?
- Error Analysis
  - Truncation Error or Discretisation Error

Chaos: *Lorenz equations*

$$\boxed{\frac{dx}{dt} = -\sigma x + \sigma y}$$

$$\boxed{\frac{dy}{dt} = rx - y - xz}$$

$$\boxed{\frac{dz}{dt} = -bz + xy}$$

- With the following parameter values $\sigma = 10, b = 2.666667$, and $r = 28$.
- Employ initial conditions of $x = y = z = 5$ and integration from $t = 0$ to 20.
- Repeat the simulation with $x = 5.001$.

Second-Order ODE

Heat Tranfer for Conduction:

$$\boxed{\frac{d^2T}{dx^2} + h'(T_a - T) = 0}$$

Boundary conditions are

$$T(0) = T_1$$
$$T(L) = T_2$$

For a 10 m rod with $T_a = 20, T_1 = 40, T_2 = 200$, and $h' = 0.01$, the solution is

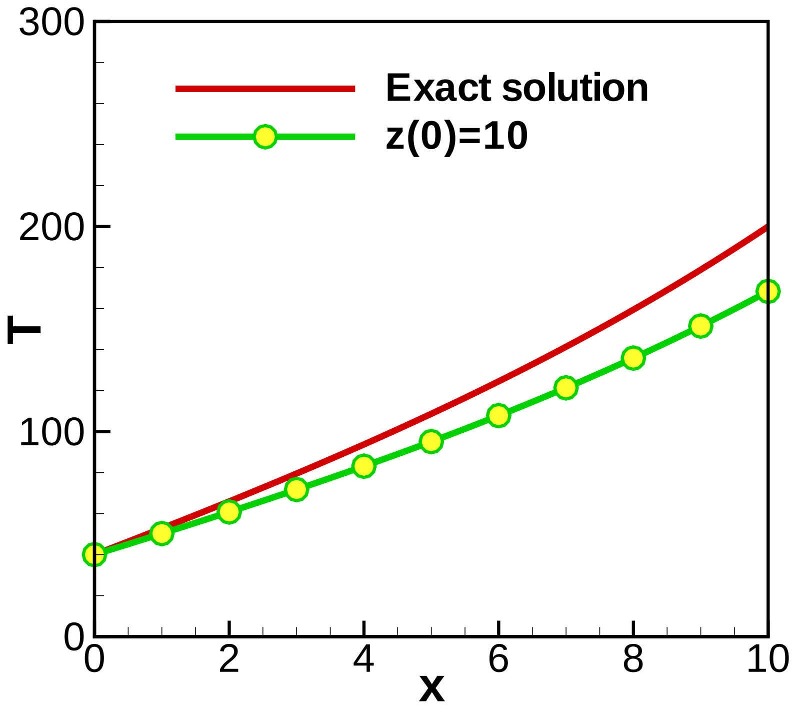
$$\boxed{T = 73.4523 \exp^{0.1x} - 53.4523 \exp^{-0.1x} + 20}$$

Second-Order ODE - Cont’d

$$\boxed{\frac{dT}{dx} = z}$$

$$\boxed{\frac{dz}{dx} = h'(T - T_a)}$$

- $z(0) = 10$
- $T(10) = 168.3787$

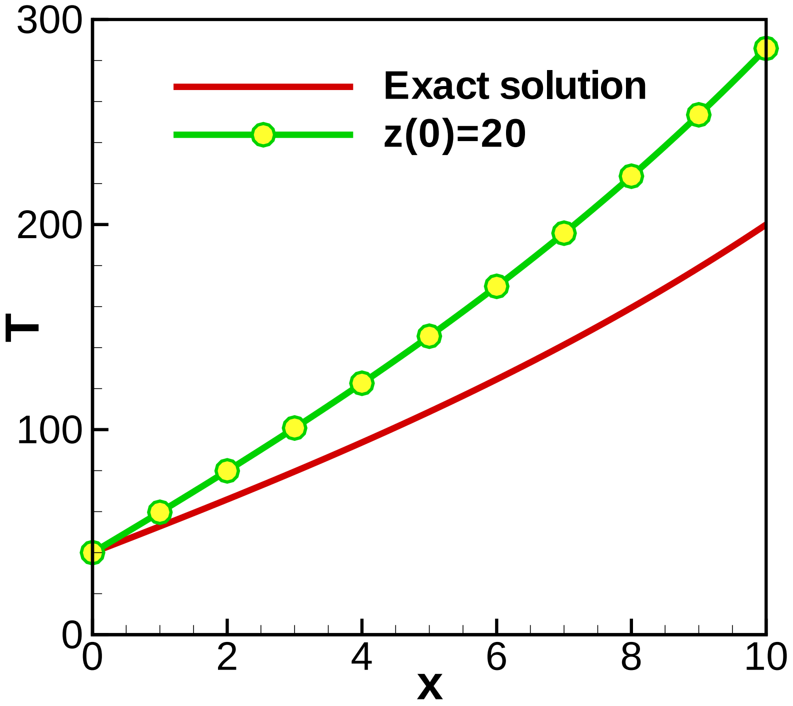


Second-Order ODE - Cont’d

$$\boxed{\frac{dT}{dx} = z}$$

$$\boxed{\frac{dz}{dx} = h'(T - T_a)}$$

- $z(0) = 20$
- $T(10) = 285.8980$



$$z(0) = 10 + \frac{20 - 10}{285.8980 - 168.3797}(200 - 168.3797) = 12.6907$$

Blasius Equation

- Blasius Equation can be written as follows:

$$\boxed{f''' + ff' = 0}$$

- The boundary conditions are
  - $\eta = 0$: $f = 0$ and $f' = 0$
  - $\eta = \infty$: $f' = 1$
- We set $f' = g$ and $f'' = q$, then

$$\boxed{(f'')' = -f(f')}$$

$$\boxed{\frac{dq}{d\eta} = -fg}$$

Blaisius Equation -Cont'd

- The modified ODE

$$\frac{df}{d\eta} = g,$$

$$\frac{dg}{d\eta} = q,$$

$$\frac{dq}{d\eta} = -fg.$$

- The new boundary conditions are
  - At $\eta = 0$, $f = 0$, $g = 0$, $q = a$.
  - At $\eta = \infty$, Check if $g = 1$.