## Quiz 1 - Computational Physics II

Date: Thursday 27 February 2025 Duration: 45 minutes Credits: 20 points (Hquestions) Type of evaluation: LAB  Provide short and concise answers to the following items:  1. (5 points) Integration methods for Ordinary Differential Equations (ODEs)  (a) Explain the difference between explicit and implicit Euler integrators for ODEs.  (b) What are the error sources when you integrate ODEs numerically in a computer?  a) Explicit Euler integrators only depends on the lost state to compute the following state while implicit taler integrators depend also in its own state to compute the actual state.  Explicit: S(t; n) = S; the F(t; S(t;)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; S(t; n)) Finite of S(t; n) = S; the F(t; n) Finite of S(t; n) = S; the F	10
Provide short and concise answers to the following items:  1. (5 points) Integration methods for Ordinary Differential Equations (ODEs)  (a) Explain the difference between explicit and implicit Euler integrators for ODEs.  (b) What are the error sources when you integrate ODEs numerically in a computer?  a) Explicit Euler integrators only depends on the lost state to compute the following state while implicit Euler integrators depend also in its own state to compute the actual state.  Explicit:  S(t; n) = Si + k F(tj, S(tj)) forficit: S(tj, n) = Si + k F(tj, S(tj)) forficit: S(tj, n) = Si + k F(tj, S(tj, n)) forficit: S(tj, n) = S	Date: Thursday 27 February 2025 Duration: 45 minutes
(a) Explain the difference between explicit and implicit Euler integrators for ODEs.  (b) What are the error sources when you integrate ODEs numerically in a computer?  a) Explicit Euler integrators only depends on the lost stake to compute the following state while implicit tuler integrators depend also in its own state to compute the actual state.  Explicit: S(tjn) = Sj th F(tj, S(tj)) for policit: S(tjn) = Sjth F(tji, S(tjn)) state (s tjn) = Sjth F(tji, S(tjn)) for the energy state (s tjn) = Sjth F(tji, S(tjn)) for the energy state (s tjn) = Sjth F(tji, S(tjn)) for the energy state (s tjn) = Sjth F(tji, S(tjn)) for the energy state (s tjn) = Sjth F(tji, S(tjn)) for the energy state (s tjn) for the en	
a) Explicit Euler integrators only depends on the lost stake to compute the tollowing state while implicit tuler integrators depend also in its own state to compute the actual state.  Explicit: S(tj.+1) = 5; th F(tj., S(ti.))   Emplicit: S(tj1) = Sj.th F(tj1, S(tj1))   Correct S(tj.+1) + Tactual state: S(tj1) + Tactual s	(a) Explain the difference between explicit and implicit Euler integrators for ODEs.
2. (5 points) Runge-Kutta methods for ODEs  (a) Explain how Runge-Kutta (RK) methods work.  (b) How do RK methods improve upon simpler integration methods like the Euler methods?  (c) Design your own third-order RK method, and write down the slopes and integrator.  (a) The RK methods comes from truncated Taylor series and it works by calculating different slopes within tj and bjtt to then made a weighted avarege from this.  (b) The advantage of RK-methods is that this method does not need calculating higher derivatives for increasing the accurrency of the solution instead it calculates the slote on intermediates points. Then it could be cheapper, computational talking, for some problems. Another advantage is that this method could be very personalizable since the weights (e.g. C1, C1, p, q) can be adapted to the problem to improve the precission, efficiency and accuracy. The stability also is higher than Euler methods since take points within thand to personalizable since the precission of the problem to improve the precission of the points within the and tight.  Stipp = Stipp + 1 (C1 K1 + 2 (2 K2 + C3 K3) -> RK method.  The shopes are:  (C1 K1 + 2 (2 K2 + C3 K3) -> RK method.  The shopes are:  (C2 K1 + 2 (2 K2 + C3 K3) -> RK method.	following state while implicit tuler integrators depend also in its own
2. (5 points) Runge-Kutta methods for ODEs  (a) Explain how Runge-Kutta (RK) methods work.  (b) How do RK methods improve upon simpler integration methods like the Euler methods?  (c) Design your own third-order RK method, and write down the slopes and integrator.  (a) The RK methods comes from truncated Taylor series and it works by calculating different slopes within to and bits to then made a weighted audrege from this.  (b) The advantage of RK methods is that this method over not need calculating higher derivatives for increasing the accurrency of the solution instead it calculates the state on intermediates points. Then it could be cheaper, computational talking, for some problems. Another advantage is that this method would be very personalizable since the weights (e.g. C1, C1, p.q.) can be adapted to the problem to improve the precision, efficiency and accuracy. The stability alls is higher tha Euler methods since take points within to and by:  S(tj) = S(tj) + 1 (C1 K1 + 2(2K2 + C3K3) -> RK method.  The slopes are:  (C1 K1 + 2(2K2 + C3K3) -> RK method.  (C2 K1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1	the method with respect to the problem?, and the machine error related with computer capacity of bus pcs.
higher derivatives for increasing the accurrency of the solution instead it calculates the state on intermediates points. Then it could be acheaper, computational talking, for some problems. Another advantage is that this mothod could be very personalizable since the weights (e.g. cr. (1, p,q) can be adapted to the problem to improve the precission, efficiency and accuracy. The stability also is higher than Euler methods since take points within transfer.  Siting = Sitis) + 1 (Cr. Kr. + 2 (2 Kr. + C3 Kr. 3) -> RM method.  The slopes are:  -0.5  Kr. = h F(tj + Rh., Sitis+ Rh) X, Kr. = h Filts+ Ph., Sitis+ Rm.)	(a) Explain how Runge-Kutta (RK) methods work.  (b) How do RK methods improve upon simpler integration methods like the Euler methods?  (c) Design your own third-order RK method, and write down the slopes and integrator.  1) The RK methods works from truncated Taylor series and it works by calculating
higher derivatives for increasing the accurrency of the solution instead it calculates the state on intermediates points. Then it could be acheaper, computational talking, for some problems. Another advantage is that this method could be very personalizable since the weights (e.g. cr. 1, p.q) can be adapted to the problem to improve the precission, efficiency and accuracy. The stability also is higher than Euler methods since take points within to and tight.  Setjen = Setjen + 1 (Cr. Kr. + 2 Cz. Kr. + C3. Kr.) -> RM method.  The slopes are:  No = h F(tj + Ph., Setjen) / Re = h F(tj + Ph., Setjen)	
olso is higher the Euler methods since take points within to and type.  Sltj. = Sltj) + 1 (C1 K1 + 2C2K2 + C3K3) -> RM method.  The slopes are:  N1 = h F(tj + Ph, Sltj + Ph) X, K2 = h F(tj + Ph, S(tj + Ph))	higher derivatives for increasing the accurrancy of the solution instead it calculates the state on intermediates points. Then it could be cheaper, computational talking, for some problems. Another advantage is that this method wild be very personalizable since the weights (e.g. ci, in, p,q) can be adapted.
SLtj+) = SLtj) + $\frac{1}{4}$ (C1 K1 + 2C2K2 + C3K3) -> RM method. The slopes are: $-0.5$ $K_1 = h F(t_j + Ph, SLt_j + Ph) \times , K_2 = h F(t_j + Ph, S(t_j + Ph))$	also is higher tha Euler methods since take points within trand byth.
M. Carrier and M. Car	SLtj+i) = SLtj) + $\frac{1}{4}$ (C1 Ki + 2C2K2 + C3K3) -> RN method. The slopes are: $-0.5$ $K_1 = h F(t_j + Ph, SLt_j + Ph) \times$ , $K_2 = h F(t_j + Ph, S(t_j + Ph))$

## 3. (5 points) ODE order reduction

Consider an object with mass, m, that falls from rest under the influence of gravity (i.e., along the Y axis). The object is also subjected to a drag force that arises from friction with air molecules, so its equation of motion reads:

$$m\frac{d^2y}{dt^2} = -m\,g + b\frac{dy}{dt}$$

where g is the acceleration of gravity and b is a friction constant.

- (a) Reduce the order of this ODE to first order and write down the resulting matrix-form equation.
- (b) Identify the slope function.
- (c) Briefly explain what the advantages of carrying out order reduction are.

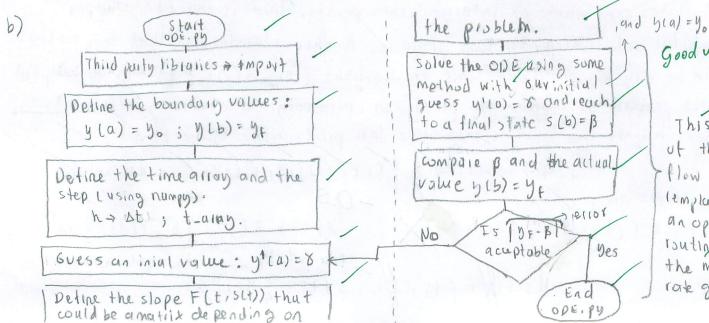
a) state vector: 
$$S(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} \not\Rightarrow F(t, S(t)) = dS(t) = \begin{bmatrix} \dot{y}(t) \\ \dot{y}(t) \end{bmatrix}$$

Operating the ODE:
$$\frac{d^2y}{dt^2} = -g + \frac{b}{m} \frac{dy}{dt} \Rightarrow \frac{dS(t)}{dt} = \begin{bmatrix} \dot{y}(t) \\ -g + \frac{b}{m} \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{y}(t) \\ \dot{y}(t) \end{bmatrix}$$

$$\Rightarrow = \begin{bmatrix} -g/y(t) \\ -g/y(t) \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{y}(t) \\ \dot{y}(t) \end{bmatrix}$$

b) The slope function F is:

- n first-order ODEs since more of the methods are optimized to solve 1st-order ODEs, which also makes the works more efficient since calculating higher order derivatives is not needed, this world be diso computationally chapper.
  - (a) Explain how the shooting method works and what it is used for.
  - (b) Sketch an algorithm workflow to implement the shooting method for an ODE in Python.
- a) The shooting method is prade to solve boundary value problems for opts, and it changes a BVP to a initial value problem (IVP) which is more easy to solve since from an initial point can be calculated all the solution. It mainly wagests on three steps: aim step, shooting step and iterative step (optimization).



This part of the workflow can be remplaced by an optimization

the more accurate guess y'(a)=8.

routine to find