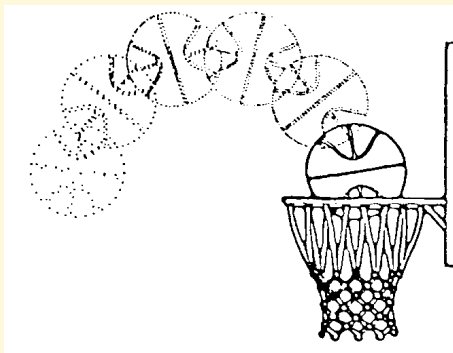




Projekt: Basketbollkast



av Grupp 3: Mikael B, Jörgen H, Sofie J, Johannes L,
Anton N, Jenny P och Rasmus W
för kursen *Beräkningsprogrammering, NUM131*

Hemsida

Försättsblad

Innehåll



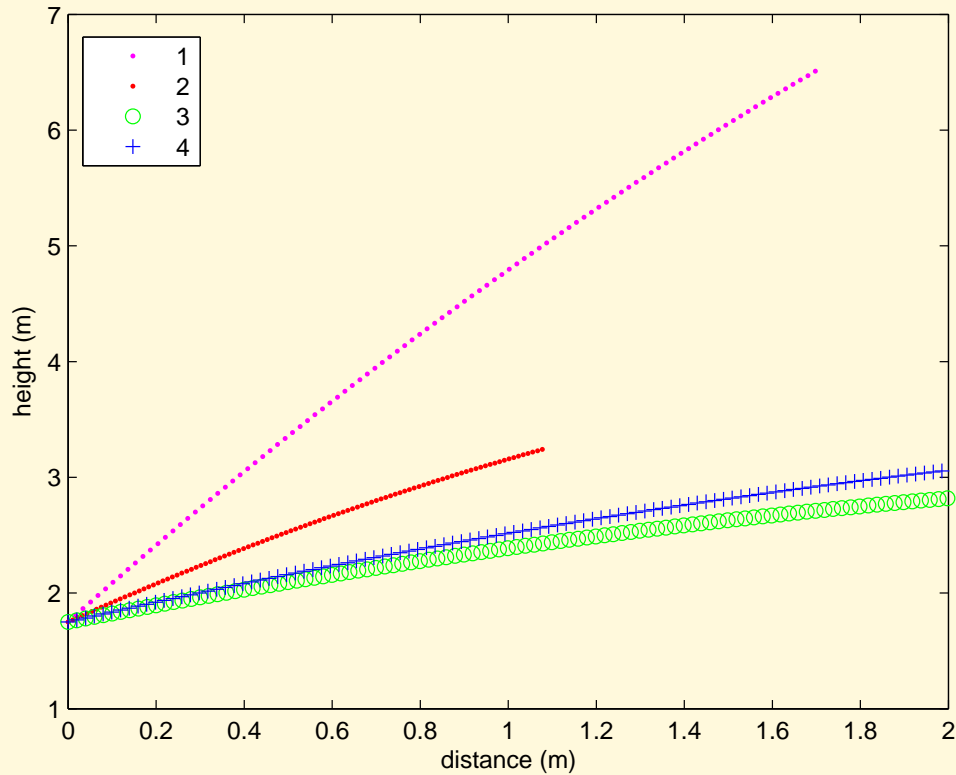
Sida 1 av 22

Tillbaka

Full Screen

Stäng

Avsluta



$$z_0 = 0.2 \text{ s}, \alpha_0 = 77^\circ, z^* = 88.264 \text{ ms}, \alpha^* = 42.868^\circ.$$



Hemsida

Försättsblad

Innehåll



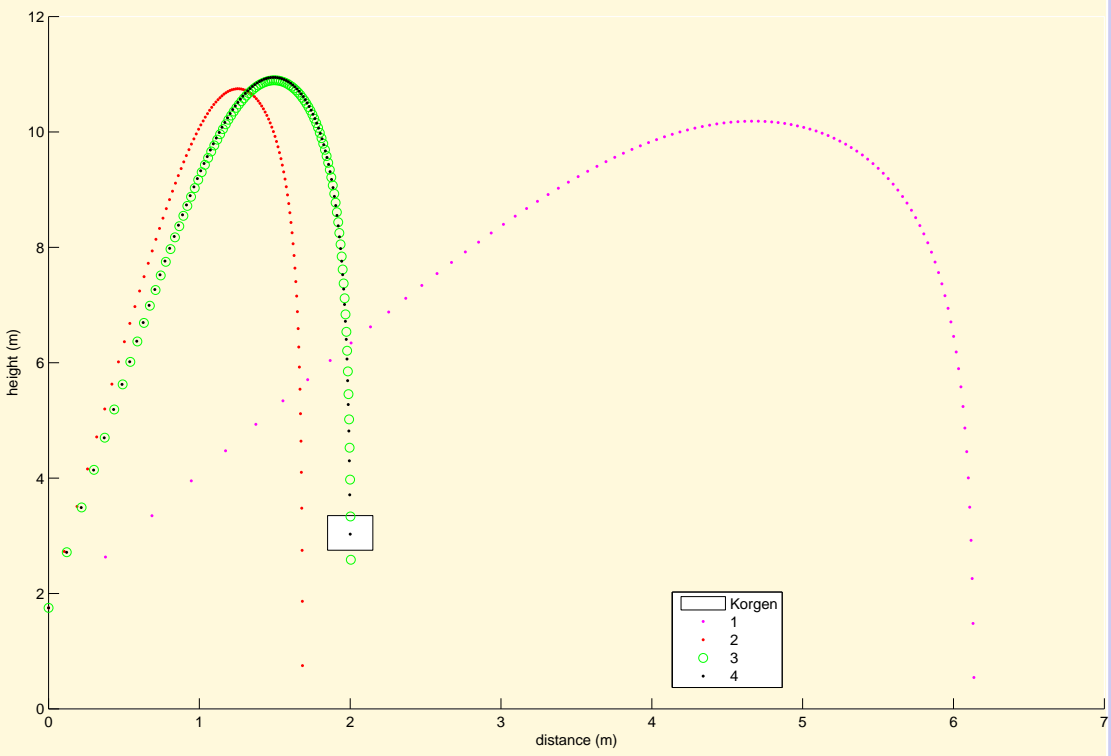
Sida 2 av 22

Tillbaka

Full Screen

Stäng

Avsluta



$$z_0 = 3.5 \text{ s}, \alpha_0 = 70^\circ, z^* = 3.554 \text{ s}, \alpha^* = 86.826^\circ.$$

Hemsida

Föreläsning

Innehåll

⏪ ⏩

◀ ▶

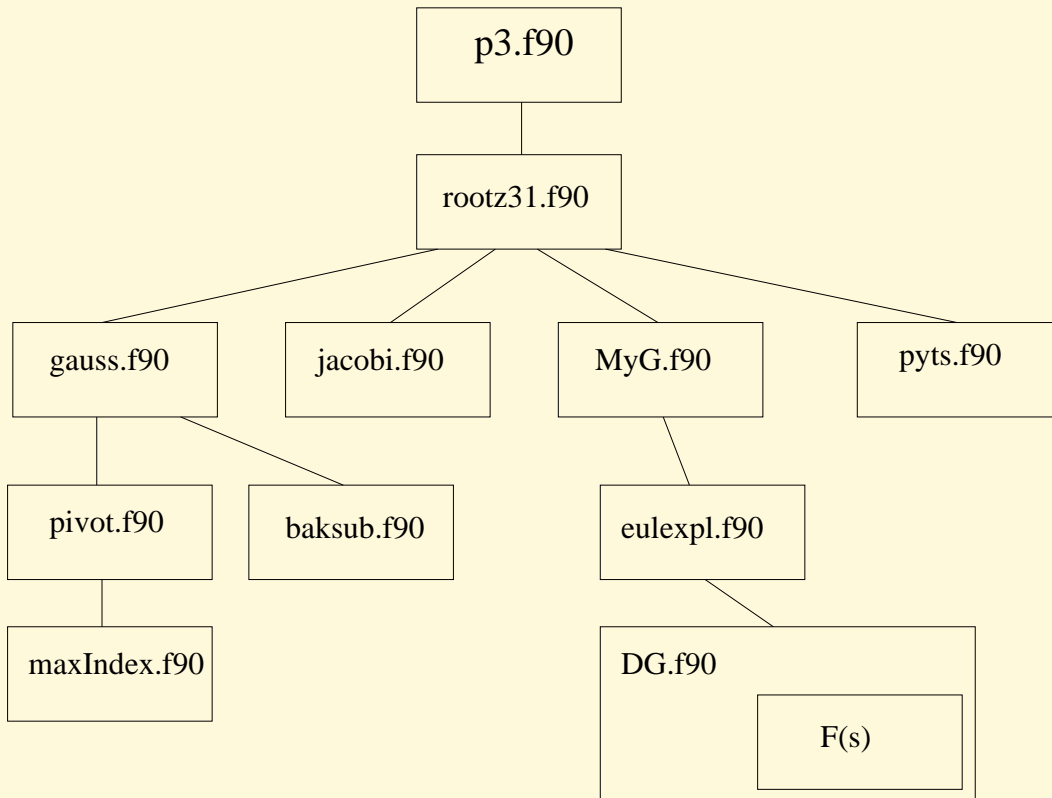
Sida 3 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Subrutiner och funktioner i programmet p3 och deras relationer.



Hemsida

Försättsblad

Innehåll



Sida 4 av 22

Tillbaka

Full Screen

Stäng

Avsluta

Interfaces:

```
subroutine rootz31(f, x0, tol, maxIter, xst, ierr)
  implicit none
  real(kind=8), intent(in)  :: x0(2), tol
  integer, intent(in)       :: maxIter
  real(kind=8), intent(out) :: xst(2)
  integer, intent(out)      :: ierr

  external f
end subroutine rootz31

real(kind = 8) function pyts(x)
  implicit none
  real(kind = 8), intent(in) :: x(2)
end function pyts

subroutine jacobi(f, fx, x, jret)
  implicit none
  real(kind = 8), intent(in) :: fx(2), x(2)
  real(kind = 8), intent(out) :: jret(2,2)
  external f
end subroutine jacobi
```



Hemsida

Föreläsning 1

Innehåll



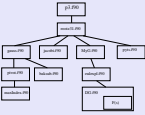
Sida 5 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Interfaces: (forts)

```
subroutine MyG(z0alpha0, xy, printing)
  implicit none
  real(kind=8), intent(in)  :: z0alpha0(2)
  real(kind=8), intent(out) :: xy(2)
  logical, intent(in)      :: printing
end subroutine MyG
```

```
subroutine eulexpl(yprim, ts, y0, divs, t, Y)
  implicit none
  real(kind=8), intent(in)  :: ts(2), y0(5)
  integer, intent(in)      :: divs
  real(kind=8), intent(out) :: t(divs+1), Y(divs+1, 5)
  external yprim
end subroutine eulexpl
```

Hemsida

Föreläsningar

Innehåll

◀▶

◀▶

Sida 6 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Interfaces: (forts)

```
subroutine DG(t, G, Gd)
  implicit none
  real(kind=8), intent(in) :: t, G(5)
  real(kind=8), intent(out) :: Gd(5)
end subroutine DG
```

```
real(kind=8) function F(s)
  implicit none
  real(kind=8), intent(in) :: s
end function F
```

Hemsida

Försätsblad

Innehåll



Sida 7 av 22

Tillbaka

Full Screen

Stäng

Avsluta

Interfaces: (forts)

```
subroutine gauss(A, b, x, r)
  implicit none
  real(kind=8), intent(in)  :: A(2,2), b(2)
  real(kind=8), intent(out) :: x(2)
  integer, intent(out)      :: r
end subroutine gauss

subroutine pivot(A, m, Aret)
  implicit none
  real(kind=8), intent(in) :: A(2, 3)
  integer, intent(in)      :: m
  real(kind=8), intent(out):: Aret(2, 3)
end subroutine pivot

integer function maxIndex(y)
  real(kind=8), intent(in) :: y(:)
end function maxIndex
```



Hemsida

Försättsblad

Innehåll



Sida 8 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Hemsida

Föreläsningar

Innehåll



Sida 9 av 22

Tillbaka

Full Screen

Stäng

Avsluta

Interfaces: (sista)

```
subroutine baksb(U, b, x, ierr)
  implicit none
  real(kind=8), intent(in)  :: U(2,2), b(2)
  real(kind=8), intent(out) :: x(2)
  integer, intent(out)      :: ierr
end subroutine baksb
```

```

p3:
write (*, *) 'Ange startvärdet z0'
read (*, *) z0
write (*, *) 'Ange begynnelsevinkeln i grader'
read (*, *) alphadeg
alpha0 = alphadeg*pi/180 ! Räknar om vinkeln från grader
                        ! till radianer
x0(1) = z0
x0(2) = alpha0 ! Sätter in startvärdena i en vektor

open(unit = 15, file='matG.m', status='unknown', &
      & action='write', iostat=oerr)
write (15, *) 'G=['
call rootz31(MyG, x0, tol, maxIter, xst, ierr)
write (15, *) '];'
close(15)
write (*, *) 'Värdena för x- och y-koordinater är&
      & skrivna i filen matG.m.'
```



Hemsida

Försättsblad

Innehåll



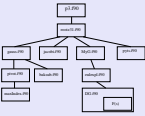
Sida 10 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Hemsida

Föreläsningar

Innehåll



Sida 11 av 22

Tillbaka

Full Screen

Stäng

Avsluta

p3: (forts)

```
z = xst(1)
```

```
alphast = xst(2) ! Får ut de slutliga värdena i form av  
! en vektor
```

```
alphastdeg = alphast*180/pi ! Räknar om vinkeln från  
! radianer till grader
```

```
write (*, *) 'Bollen måste skickas iväg med vinkeln', &  
& alphastdeg, 'grader för att träffa korgen'
```

```
write (*, *) 'Tiden det tar för bollen att nå korgen &  
&är', z, 'sekunder.'
```

```
subroutine rootz31(f, x0, tol, maxIter, xst, ierr):
```

```
  tol2 = tol**2
```

```
  x = x0
```

```
  do i = 1, maxIter
```

```
    call f(x, fx, printing)
```

```
    call jacobi(f, fx, x, jx)
```

```
    call gauss(jx, fx, deltax, r)
```

```
    x = x - deltax
```

```
    if (pyts(deltax) < tol2) then
```

```
      ierr = 0
```

```
      xst = x
```

```
      return
```

```
    end if
```

```
  end do
```

```
  ierr = 1
```



Hemsida

Föreläsningar

Innehåll



Sida 12 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Hemsida

Föreläsning

Innehåll



Sida 13 av 22

Tillbaka

Full Screen

Stäng

Avsluta

```
real(kind = 8) function pyts(x):  
real(kind = 8), intent(in):: x(2)
```

```
pyts = x(1)*x(1) + x(2)*x(2)
```



Hemsida

Föreläsning

Innehåll



Sida 14 av 22

Tillbaka

Full Screen

Stäng

Avsluta

subroutine jacobi(f, fx, x, J):

$E(1,1) = 1$

$E(1,2) = 0$

$E(2,1) = 0$

$E(2,2) = 1$

do i = 1, n

 xplushe = x + h*E(:, i)

 call f(xplushe, fxplushe, printing)

 J(:, i) = (fxplushe - fx)/h

end do

subroutine MyG(z0alpha0, xy, printing):

```
z0 = z0alpha0(1)
alpha0 = z0alpha0(2)
G0(1) = x0
G0(2) = y0
G0(3) = z0
G0(4) = s0*cos(alpha0); ! vx0, alpha0 from ground plane
G0(5) = s0*sin(alpha0); ! vy0
ts = (/ 0, 1 /)
call eulexpl(DG, ts, G0, divs, t, G)
if (printing) then
  do i = 1, size(G, 1)
    write (unit=15, fmt='(f20.14)', advance='no') &
      & G(i, 1)
    if (mod(i, 3) == 0) then
      write (15, *) ' ...'
    end if
  end do
  write (15, *) ';'
```



Hemsida

Föreläsning

Innehåll



Sida 15 av 22

Tillbaka

Full Screen

Stäng

Avsluta



subroutine MyG(z0alpha0, xy, printing): (forts)

```
do i = 1, size(G, 1)
  write (unit=15, fmt='(f20.14)', advance='no') &
    & G(i, 2)
  if (mod(i, 3) == 0) then
    write (15, *) ' ...'
  end if
end do
write (15, *) ';'
```

end if

```
xy(1) = G(size(G, 1), 1) - x_B
xy(2) = G(size(G, 1), 2) - y_B
```

Hemsida

Försättsblad

Innehåll



Sida 16 av 22

Tillbaka

Full Screen

Stäng

Avsluta



subroutine eulexpl(yprim, ts, y0, divs, tret, Yret):

h = (ts(2) - ts(1))/divs ! h är steglängden

! skapar en vektor t med tidpunkter

do i = 1, divs+1

 tret(i) = (i-1)*h+ts(1)

end do

Yret(1, :) = y0 !tilldelning av initialvärde

do i = 1, divs

 Ya = Yret(i,:) !(a = aktuell)

 call yprim(tret(i), Ya, Yd)

 Yret(i+1, :) = Yret(i, :) + Yd*h

end do

Hemsida

Försättsblad

Innehåll



Sida 17 av 22

Tillbaka

Full Screen

Stäng

Avsluta

subroutine DG(t, G, Gd):

$$x = G(1)$$

$$y = G(2)$$

$$z = G(3)$$

$$vx = G(4)$$

$$vy = G(5)$$

$$xd = vx * z$$

$$yd = vy * z$$

$$s = \text{sqrt}(xd**2 + yd**2)$$

$$\text{alpha} = \text{acos}(xd / s)$$

$$! \text{ Gd} = 0$$

$$\text{Gd}(1) = xd$$

$$\text{Gd}(2) = yd$$

$$\text{Gd}(3) = 0$$

$$\text{Gd}(4) = -F(s) * z * \text{cos}(\text{alpha})$$

$$\text{Gd}(5) = -F(s) * z * \text{sin}(\text{alpha}) - \text{gravity}$$

contains



Hemsida

Föreläsning

Innehåll



Sida 18 av 22

Tillbaka

Full Screen

Stäng

Avsluta



Hemsida

Förelätsblad

Innehåll



Sida 19 av 22

Tillbaka

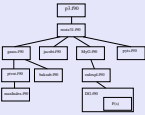
Full Screen

Stäng

Avsluta

real(kind=8) function F(s):

$$F = (1.0/2.0) * \rho * c_w * (\pi/4) * d**2 * s**2$$



Hemsida

Försätsblad

Innehåll



Sida 20 av 22

Tillbaka

Full Screen

Stäng

Avsluta

subroutine gauss(A, b, x, r):

```
det = A(1,1)*A(2,2) - A(1,2)*A(2,1)
```

```
if(abs(det) < tol) then
```

```
    stop 'Fel i funktionen gauss. Singulär matris'
```

```
end if
```

```
x(1) = (b(1)*A(2,2) - b(2)*A(1,2))/det
```

```
x(2) = (b(2)*A(1,1) - b(1)*A(2,1))/det
```

```
r = 2
```



Genomförandeplan

- Fungerande MATLAB-prototyp
- Översättning till Fortran 95
 - Konstruktion av interfaces
- Skriva Fortran-program-, subrutin- och funktionsinterfaces
- Konstruktion av skelett av interfaces
- Test av varje enskilt program & subrutin
- Hopsättning av fungerande skelett + test
- MATLAB för att visa resultat

Hemsida

Försättsblad

Innehåll



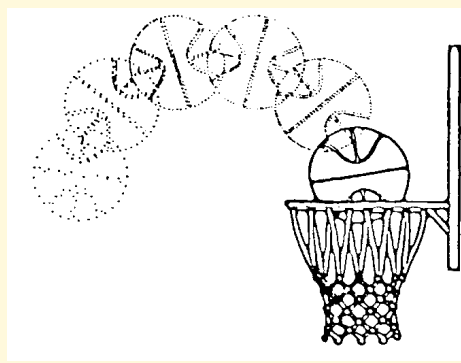
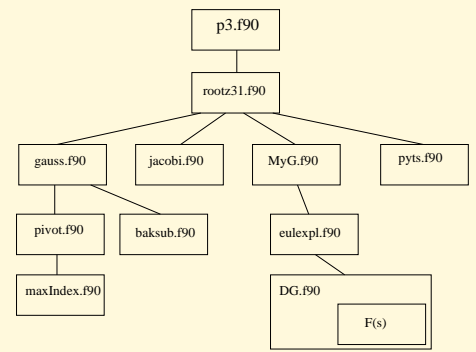
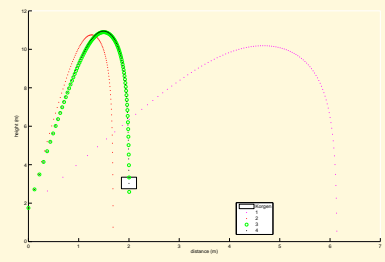
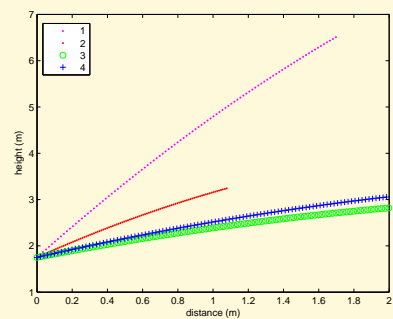
Sida 21 av 22

Tillbaka

Full Screen

Stäng

Avsluta

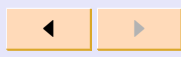
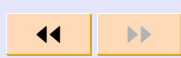


Trevlig sommar!

Hemsida

Försätsblad

Innehåll



Sida 22 av 22

Tillbaka

Full Screen

Stäng

Avsluta