

PROGRAMMING FOR COMPUTATIONAL CHEMISTRY

Subroutines and functions

Emanuele Coccia

Dipartimento di Scienze Chimiche e Farmaceutiche

Top-down design

- **Decomposition** of the main algorithm
- Logical subdivisions

Top-down design

- **Decomposition** of the main algorithm
- Logical subdivisions
- Two Fortran procedures: **subroutine** and **function**

Top-down design

- **Decomposition** of the main algorithm
- Logical subdivisions
- Two Fortran procedures: **subroutine** and **function**
- Subtasks easy to test and debug

Top-down design

- **Decomposition** of the main algorithm
- Logical subdivisions
- Two Fortran procedures: **subroutine** and **function**
- Subtasks easy to test and debug
- Reusable

Top-down design

- **Decomposition** of the main algorithm
- Logical subdivisions
- Two Fortran procedures: **subroutine** and **function**
- Subtasks easy to test and debug
- Reusable
- Easy to read and maintain

- Fortran procedure:
 *subroutine name(list of **formal** arguments)*
 declaration
 instructions
 ...
 return
end subroutine name

- Fortran procedure:
 *subroutine name(list of **formal** arguments)*
 declaration
 instructions
 ...
 return
 end subroutine name
- In the **calling** program: *call name(list of **current** arguments)*

Subroutines

- Fortran procedure:
*subroutine name(list of **formal** arguments)*
declaration
instructions
...
return
end subroutine name
- In the **calling** program: *call name(list of **current** arguments)*
- Consistency in number, order and type between **current** and **formal** arguments

- Fortran procedure:
*subroutine name(list of **formal** arguments)*
declaration
instructions
...
return
end subroutine name
- In the **calling** program: *call name(list of **current** arguments)*
- Consistency in number, order and type between **current** and **formal** arguments
- Can return **multiple** results

- Dummy arguments:

Subroutines

- Dummy arguments:
 - Used to **pass data** and not changed

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them

Subroutines

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute

Subroutines

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute
- **Pass-by-reference** scheme

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute
- **Pass-by-reference** scheme
- Local variables

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute
- **Pass-by-reference** scheme
- Local variables
- Array with its dimension as arguments (**explicit-shape dummy array**)

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute
- **Pass-by-reference** scheme
- Local variables
- Array with its dimension as arguments (**explicit-shape dummy array**)
- Example **area_triangolo.f90**

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute
- **Pass-by-reference** scheme
- Local variables
- Array with its dimension as arguments (**explicit-shape dummy array**)
- Example **area_triangolo.f90**
- Example **array_subroutine.f90**

Subroutines

- Dummy arguments:
 - Used to **pass data** and not changed
 - Used as **output** variable
 - no memory allocated for them
- **Intent** attribute
- **Pass-by-reference** scheme
- Local variables
- Array with its dimension as arguments (**explicit-shape dummy array**)
- Example **area_triangolo.f90**
- Example **array_subroutine.f90**
- **Save attribute** for local variables

- Fortran procedure:
 (type) function name(formal arguments)
 declaration
 instructions
 ...
 return
 end function name
- Only one value returned

Functions

- Fortran procedure:
 (type) function name(formal arguments)
 declaration
 instructions
 ...
 return
 end function name
- Only one value returned
- Name of return variable = function name
- Type of return variable = function type

Functions

- Fortran procedure:
 (type) function name(formal arguments)
 declaration
 instructions
 ...
 return
 end function name
- Only one value returned
- Name of return variable = function name
- Type of return variable = function type
- Same passing scheme of subroutines

- Fortran procedure:
(type) function name(formal arguments)
declaration
instructions
...
return
end function name
- Only one value returned
- Name of return variable = function name
- Type of return variable = function type
- Same passing scheme of subroutines
- Example `distance.f90`

Examples

Functions:

- `norma_matrice.f90`

Examples

Functions:

- `norma_matrice.f90`
- `quadratura_trapezio.f90`

Functions:

- `norma_matrice.f90`
- `quadratura_trapezio.f90`
- `bisezione.f90`

Functions:

- `norma_matrice.f90`
- `quadratura_trapezio.f90`
- `bisezione.f90`

Subroutines:

- `root_eq2.f90`

Functions:

- `norma_matrice.f90`
- `quadratura_trapezio.f90`
- `bisezione.f90`

Subroutines:

- `root_eq2.f90`
- `bubble_sort.f90`

Functions:

- `norma_matrice.f90`
- `quadratura_trapezio.f90`
- `bisezione.f90`

Subroutines:

- `root_eq2.f90`
- `bubble_sort.f90`
- `pi_montecarlo.f90`

Functions:

- `norma_matrice.f90`
- `quadratura_trapezio.f90`
- `bisezione.f90`

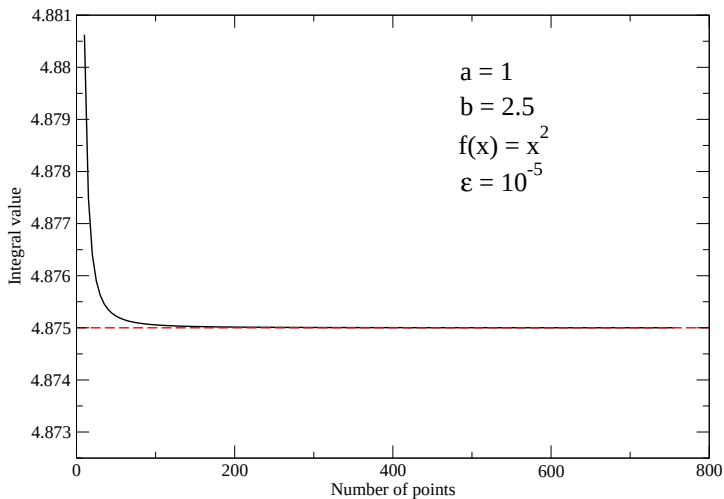
Subroutines:

- `root_eq2.f90`
- `bubble_sort.f90`
- `pi_montecarlo.f90`

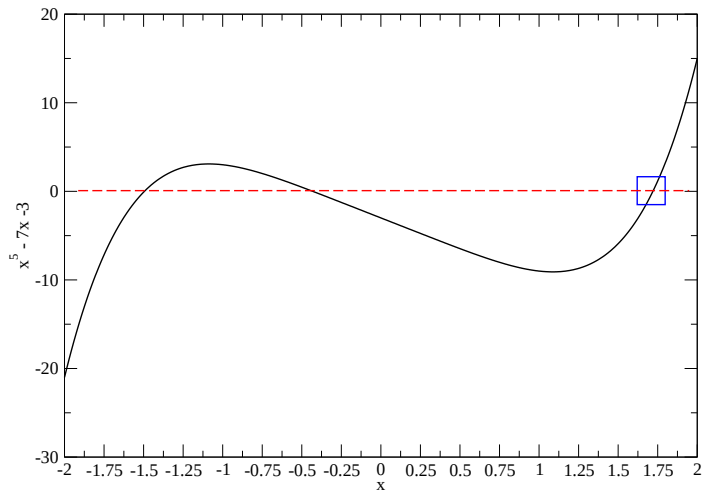
Subroutines/functions:

- `media.f90`

Integral $f(x) = x^2$



$$f(x) = x^5 - 7x - 3$$



Bubble sort

3 -1 2 4 0

Bubble sort

3 -1 2 4 0
-1 3 2 4 0 |

Bubble sort

3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |

Bubble sort

3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |

Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****
```

Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****
-1 2 3 0 | 4
```


Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****
-1 2 3 0 | 4
-1 2 3 0 | 4
```

Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****

-1 2 3 0 | 4
-1 2 3 0 | 4
-1 2 0 3 | 4
*****
```

Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****

-1 2 3 0 | 4
-1 2 3 0 | 4
-1 2 0 3 | 4
*****

-1 2 0 | 3 4
```

Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****

-1 2 3 0 | 4
-1 2 3 0 | 4
-1 2 0 3 | 4
*****

-1 2 0 | 3 4
-1 0 2 | 3 4
*****
```

Bubble sort

```
3 -1 2 4 0
-1 3 2 4 0 |
-1 2 3 4 0 |
-1 2 3 4 0 |
-1 2 3 0 4 |
*****

-1 2 3 0 | 4
-1 2 3 0 | 4
-1 2 0 3 | 4
*****

-1 2 0 | 3 4
-1 0 2 | 3 4
*****

-1 0 | 2 3 4
```