Examples: single variable functions

Example 1

- Initialize DACE to perform 20-th order computations

```
DA::init( 20, 1 );
```

- Initialize x as DA number

```
DA x = DA(1);
```

- Compute $y = \sin(x)$

```
DA y = sin(x);
```

- Print to screen

```
cout << "x" << endl << x << endl;
cout << "sin(x)" << endl << sin(x);
```

Examples: single variable functions

- Example 1
  - Compare result with:

$$\mathcal{T}_{\sin(x)} = \sum_{i=0}^{\infty} a_i\, x^i = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

$$\Downarrow$$

$$\begin{aligned}
a_0 &= a_2 = a_4 = \cdots = 0 \\
a_1 &= 1 \\
a_3 &= -1/6 = -0.16666\ldots \\
a_5 &= 1/120 = 0.0083333\ldots \\
a_7 &= -1/5040 = 0.00019841269841 \\
&\ldots
\end{aligned}$$

Examples: single variable functions

- Example 2
  - Initialize x as DA number
  - Compute $\sin^2(x)$
  - Compute $\cos^2(x)$
  - Compute $\sin^2(x) + \cos^2(x)$

$$\boxed{\text{Verify that } \sin^2(x) + \cos^2(x) = 1}$$

Examples: single variable functions

- Example 3: Taylor expansion about $x_0 \neq 0$
  - Consider $f(x) = \dfrac{1}{x + 1/x}$ $\Rightarrow$ $f'(x) = \dfrac{1/x^2 - 1}{(x + 1/x)^2}$
  - Compute f and f' at 3 analytically: $f(3) = \dfrac{3}{10} = 0.3$

    $$f'(3) = -\frac{2}{25} = -0.08$$
  - Compute $f(3)$ and $f'(3)$ with DA:
    - Initialize DACE for 1st order computations: `DA::init( 1, 1 );`
    - Initialize x as a DA number around 3: `DA x = 3 + DA(1);`
    - Evaluate only $f(x)$
  - Repeat for different orders

Examples: single variable functions

- Example 4:
  - Initialize DACE to perform 20th-order computations in 1 variable
  - Compute $\cos(x) - 1$ and print it
  - Compute $[\cos(x) - 1]^{11}$ $\Rightarrow$ `pow(cos(x)-1,11)`
  - Compute $[\cos(x) - 1]^{n}$ for $n = 1, \ldots, 10$

Examples: single variable functions

- Example 5: derivation and integration
  - Initialize DACE to perform 20-th order computations in 1 variable
  - Compute $\sin(x)$
  - Compute $\mathrm{d}\sin(x)/\mathrm{d}x$ `sin(x).deriv(1)`
  - Verify that it is equal to $\cos(x)$ (find the difference and explain)
  - Compute $\int \sin(x)\,\mathrm{d}x$ `sin(x).integ(1)`
  - Verify that it is equal to $-\cos(x)$ (find the difference and explain)

Examples: single variable functions

- Example 6: definite integral
  - Given $f(x)$ and its indefinite integral $F(x) = \int f(x)\mathrm{d}x$

    $$\Rightarrow \int_a^b f(x)\mathrm{d}x = F(b) - F(a)$$

  - Use DA to compute $\int_{-1}^{1} \mathrm{erf}(x)$:

```
const double pi = 4.0*atan(1.0);
```

    - Initialize DACE for 24-th order computations
    - Compute the Taylor expansion of $\mathrm{erf}(x) = \exp(-x^2/2)/\sqrt{(2\pi)}$
    - Compute the Taylor expansion of $F = \int \mathrm{erf}(x)$
    - Compute $F(1) - F(-1)$ $\Rightarrow$ `F.evalScalar(1)-F.evalScalar(-1)`
    - Verify that the result is 0.682689492137

Examples: Multivariate Functions

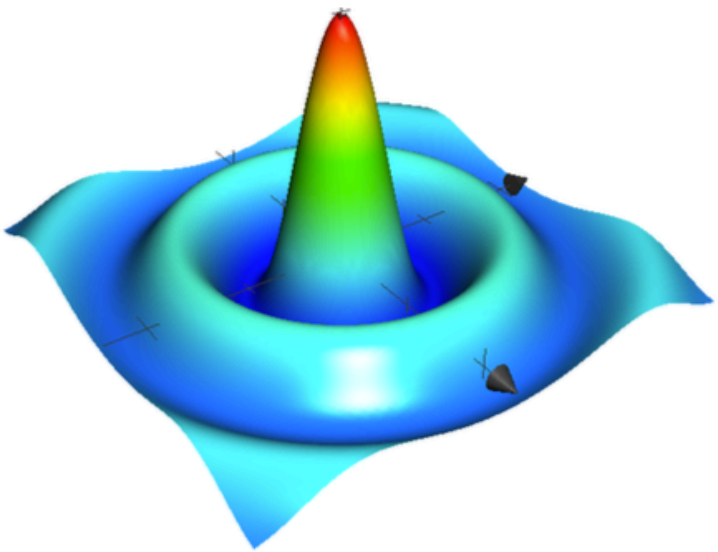
- Example 7: sombrero function
  - Initialize DACE to perform 10th-order computations in 2 variables

```
DA::init( 10, 2 );
```

  - Initialize $\boldsymbol{x}$ as a two-dimensional vector of DA numbers (AlgebraicVector) around the point $(x_1, x_2) = (0, 0)$:

```
AlgebraicVector<DA> x(2);
x[0] = DA(1);
x[1] = DA(2);
```

  - Print $\boldsymbol{x}$ to screen
  - Compute $z = \dfrac{\sin(\sqrt{(x_1^2 + x_2^2)})}{\sqrt{(x_1^2 + x_2^2)}}$

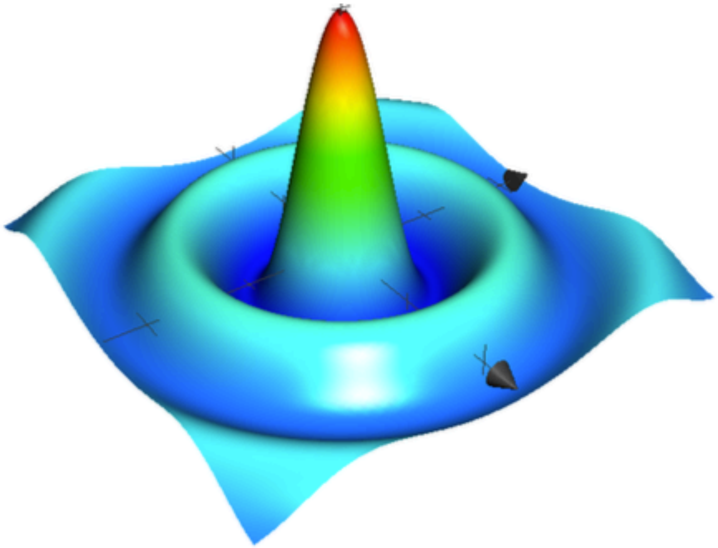


Examples: Multivariate Functions

- Example 7: sombrero function
  - Initialize $\boldsymbol{x}$ as a two-dimensional vector of DA numbers (AlgebraicVector) around the point $(x_1, x_2) = (2, 3)$:

```
AlgebraicVector<DA> x(2);
x[0] = 2 + DA(1);
x[1] = 3 + DA(2);
```

  - Compute $z = \dfrac{\sin(\sqrt{(x_1^2 + x_2^2)})}{\sqrt{(x_1^2 + x_2^2)}}$



Examples: Multivariate Functions

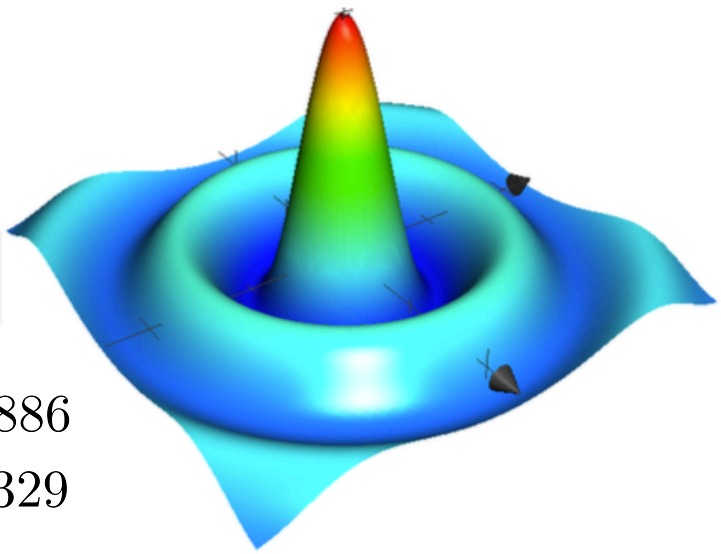
- Example 8: gradient of the sombrero function
  - Initialize DACE to perform 1st-order computations in 2 variables

```
DA::init( 1, 2 );
```

  - Compute the Taylor expansion of the sombrero function around $(x_1, x_2) = (2, 3)$
  - Compute the gradient of the sombrero function at $(x_1, x_2) = (2, 3)$:

$$z = \frac{\sin(\sqrt{(x_1^2 + x_2^2)})}{\sqrt{(x_1^2 + x_2^2)}} \Rightarrow$$

```
z.gradient()
```



  - Verify it is equal to:

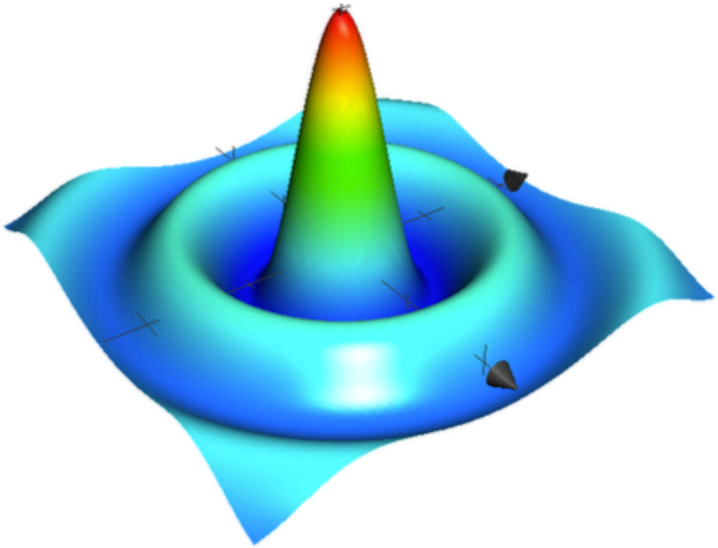
$$\partial z / \partial x = -0.1184886$$
$$\partial z / \partial y = -0.1777329$$

Examples: Multivariate Functions

- Example 8: gradient of the sombrero function
  - Initialize DACE to perform 5th-order computations in 2 variables

    ```
    DA::init( 5, 2 );
    ```

  - Compute the Taylor expansion of the sombrero function around $(x_1, x_2) = (2, 3)$
  - Compute the gradient of the sombrero function around $(x_1, x_2) = (2, 3)$
  - Print the result to screen



Advanced: Inversion of Polynomials

- Note: Assume we are performing 1st-order computations

matrix

$$\boldsymbol{y} = \boldsymbol{y}_0 + M\,\delta\boldsymbol{x}$$

- Invert the 1st-order polynomial:
  - Step 1: $\boldsymbol{y} - \boldsymbol{y}_0 = M\,\delta\boldsymbol{x} \Rightarrow \delta\boldsymbol{y} = M\,\delta\boldsymbol{x}$
  - Step 2: $\delta\boldsymbol{x} = M^{-1}\,\delta\boldsymbol{y}$

Advanced: Inversion of Polynomials

- Note: Assume we are performing 1st-order computations

matrix

$$\boldsymbol{y} = \boldsymbol{y}_0 + M\,\delta\boldsymbol{x}$$

- Invert the 1st-order polynomial:
  - Step 1: $\boxed{\boldsymbol{y} - \boldsymbol{y}_0 = M\,\delta\boldsymbol{x}} \Rightarrow \delta\boldsymbol{y} = M\,\delta\boldsymbol{x}$
  - Step 2: $\delta\boldsymbol{x} = M^{-1}\,\delta\boldsymbol{y}$

$\Rightarrow$ We first need to subtract the constant part

- The same applies for high-order polynomials:

$$\boldsymbol{y} = \mathcal{T}_{\boldsymbol{y}}(\delta\boldsymbol{x}) \Rightarrow \boldsymbol{y} - \boldsymbol{y}_0 = \delta\boldsymbol{y} = \tilde{\mathcal{T}}_{\delta\boldsymbol{y}}(\delta\boldsymbol{x}) \Rightarrow \delta\boldsymbol{x} = \tilde{\mathcal{T}}_{\delta\boldsymbol{y}}^{-1}(\delta\boldsymbol{y})$$

Advanced: Inversion of Polynomials

- Example 9:
  - Initialize DACE to perform 10th-order computations in 1 variable

    ```
    DA::init( 10, 1 );
    ```

  - Initialize x as a DA number around 0
  - Compute the Taylor expansion of $\sin(x)$ and store it in a 1-dimensional DA vector y
  - Invert the polynomial ⇨ `y.invert()`
  - Verify that it is equal to $\arcsin(x)$
  - Do the same for $\cos(x)$ around 0

Advanced: Inversion of Polynomials

- Example 10: Map inversion (transf. cylindric to Cartesian)
  - Initialize DACE to perform 10th-order computations in 3 variables
  - Initialize (r, θ, z) as a DA vector $\boldsymbol{cyl}$ around $(100, 0, 0)$:

```
AlgebraicVector<DA> cyl(3);
cyl[0] = 100.0+DA(1);
cyl[1] = DA(2)*pi/180.0;
cyl[2] = DA(3);
```

  - Initialize (x, y, z) as a DA vector $\boldsymbol{cart}$ and compute:

$$\begin{cases} x &= r\ \cos\theta \\ y &= r\ \sin\theta \\ z &= z \end{cases} \quad \Longrightarrow \quad [x, y, z] = \mathcal{T}_{cyl2cart}(\delta r, \delta\theta, \delta z)$$

Advanced: Inversion of Polynomials

- Example 10: Map inversion (transf. cylindric to Cartesian)
  - Subtract the constant part:

$$[\delta x, \delta y, \delta z] = \tilde{\mathcal{T}}_{cyl2cart}(\delta r, \delta\theta, \delta z) \qquad \text{DirMap}$$

  - Invert the map:

$$[\delta r, \delta\theta, \delta z] = \tilde{\mathcal{T}}^{-1}_{cyl2cart}(\delta x, \delta y, \delta z) \qquad \text{InvMap}$$

  - How can we check the result?

Advanced: Inversion of Polynomials

- Example 10: Map inversion (transf. cylindric to Cartesian)
  - Subtract the constant part:

$$[\delta x, \delta y, \delta z] = \tilde{\mathcal{T}}_{cyl2cart}(\delta r, \delta\theta, \delta z) \qquad \text{DirMap}$$

  - Invert the map:

$$[\delta r, \delta\theta, \delta z] = \tilde{\mathcal{T}}^{-1}_{cyl2cart}(\delta x, \delta y, \delta z) \qquad \text{InvMap}$$

  - How can we check the result? Verify that:

$$\tilde{\mathcal{T}}_{cyl2cart} \circ \tilde{\mathcal{T}}^{-1}_{cyl2cart} = \mathcal{I}$$

Advanced: Inversion of Polynomials

- Example 10: Map inversion (transf. cylindrical to Cartesian)
  - Subtract the constant part:

$$[\delta x, \delta y, \delta z] = \tilde{\mathcal{T}}_{cyl2cart}(\delta r, \delta\theta, \delta z) \qquad \text{DirMap}$$

  - Invert the map:

$$[\delta r, \delta\theta, \delta z] = \tilde{\mathcal{T}}^{-1}_{cyl2cart}(\delta x, \delta y, \delta z) \qquad \text{InvMap}$$

  - How can we check the result? Verify that:

$$\tilde{\mathcal{T}}_{cyl2cart} \circ \tilde{\mathcal{T}}^{-1}_{cyl2cart} = \mathcal{I} \quad \Rightarrow \quad \tilde{\mathcal{T}}_{cyl2cart}(\tilde{\mathcal{T}}^{-1}_{cyl2cart}) = [\delta x, \delta y, \delta z]$$

```
DirMap.eval(InvMap)
```

Advanced: Integration of ODEs

- Example 11: Integration of 2-body dynamics
  - Implement the 2-body dynamics:

```
template<typename T> AlgebraicVector<T> TBP( AlgebraicVector<T> x, double t )
{
    AlgebraicVector<T> pos(3), res(6);
    pos[0] = x[0]; pos[1] = x[1]; pos[2] = x[2];
    T r = pos.vnorm();
    const double mu = 398600; // km^3/s^2
    res[0] = x[3]; res[1] = x[4]; res[2] = x[5];
    res[3] = -mu*pos[0]/(r*r*r);
    res[4] = -mu*pos[1]/(r*r*r);
    res[5] = -mu*pos[2]/(r*r*r);
    return res;
}
```

Advanced: Integration of ODEs

- Example 11: Integration of 2-body dynamics
  - Implement an Euler integrator:

```
template<typename T> AlgebraicVector<T> euler( AlgebraicVector<T> x, double t0,
double t1)
{
        const double hmax = 0.1;
        int steps = ceil((t1-t0)/hmax);
        double h = (t1-t0)/steps;
        double t = t0;

        for( int i = 0; i < steps; i++ )
        {
                x = x + h*TBP(x,t);
                t += h;
        }
    return x;
}
```

Advanced: Integration of ODEs

- Example 11: Integration of 2-body dynamics
  - Set initial conditions for the pericenter of an orbit with $e = 0.5$:

```
AlgebraicVector<double> x0(6);
const double mu = 398600;
const double ecc = 0.5;
x0[0] = 6678.0; // 300 km altitude
x0[1] = 0.0;
x0[2] = 0.0;
x0[3] = 0.0;
x0[4] = sqrt(mu/6678.0)*sqrt(1+ecc);
x0[5] = 0.0;
```

  - Integrate for half the orbital period

```
const double pi = 4.0*atan(1.0);
double a = 6678.0/(1-ecc);
xf = euler( x0, 0.0, pi*sqrt(a*a*a/mu));
```

Advanced: Integration of ODEs

- Example 11: Integration of 2-body dynamics
  - Initialize DACE to perform 1st-order computations in 6 variables

```
DA::init( 1, 6 );
```

  - Set the initial conditions as a DA vector:

```
AlgebraicVector<DA> x0(6);
const double mu = 398600;
const double ecc = 0.5;
x0[0] = 6678.0 + DA(1); // 300 km altitude
x0[1] = 0.0 + DA(2);
x0[2] = 0.0 + DA(3);
x0[3] = 0.0 + DA(4);
x0[4] = sqrt(mu/6678.0)*sqrt(1+ecc) + DA(5);
x0[5] = 0.0 + DA(6);
```

  - Integrate for half the orbital period

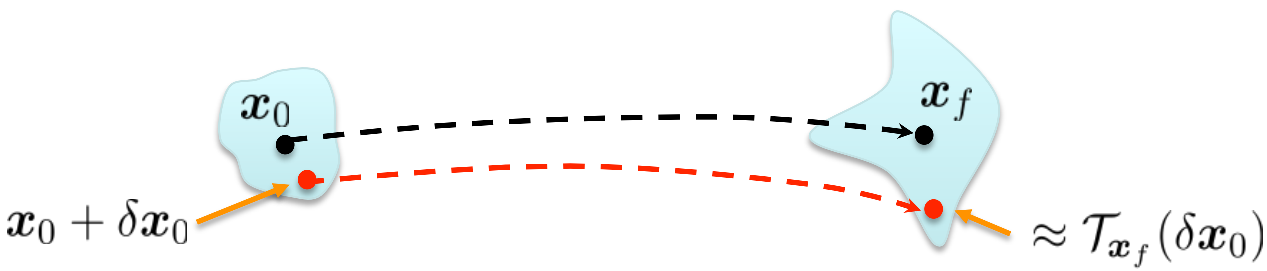
```
xf = euler( x0, 0.0, pi*sqrt(a*a*a/mu));
```

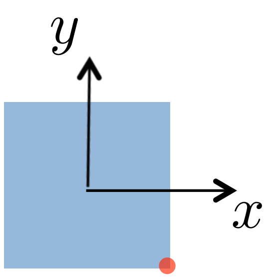
Advanced: Integration of ODEs

- Example 11: Integration of 2-body dynamics

```
xf = euler( x0, 0.0, pi*sqrt(a*a*a/mu));
```

  - We obtain: $[\boldsymbol{x}_f] = \mathcal{T}(\delta \boldsymbol{x}_0)$
  - Evaluate the polynomial for a displaced initial condition:






```
AlgebraicVector<DA> Deltax0(6);

Deltax0[0] =  1.0; // km
Deltax0[1] = -1.0;
Deltax0[2] =  0.0;
...
Deltax0[5] = 0.0;
```

```
xf.eval(Deltax0)
```

Advanced: Parametric Implicit Equations

- Example 12: Kepler's equation

$$f(E) = E - e\sin E - \sqrt{\frac{\mu}{a^3}}(t - t_0) = 0$$

- Implements Newton's method to find reference solution $\bar{E}$ for $\bar{a}$ and $\bar{e}$

$$\boxed{E_k = E_{k-1} - \frac{f(E_{k-1})}{f'(E_{k-1})}}$$

```
const double mu = 1.0;
DA a = 1.0;
DA e = 0.5;
double t = pi/2.0;
DA M = sqrt(mu/(a*a*a))*t; //real at this stage

DA EccAn = M; //first guess

double err = abs(EccAn - e*sin(EccAn) - M);

while(err>tol)
{
    EccAn = EccAn — (EccAn - e*sin(EccAn) — M)/(1 - e*cos(EccAn));
    err = abs(EccAn - e*sin(EccAn) - M);
}
```

Advanced: Parametric Implicit Equations

- Example 12: Kepler's equation

$$f(E) = E - e\sin E - \sqrt{\frac{\mu}{a^3}}(t - t_0) = 0$$

  - Initialize a and e as DA numbers around the reference
  - Perform again the Newton's steps

```
a = 1.0 + DA(1);
e = 0.5 + DA(2);
M = sqrt(mu/(a*a*a))*t; // now M is a DA
int i = 1;
while( i <= 10)
{
        EccAn = EccAn – (EccAn - e*sin(EccAn) – M)/(1 - e*cos(EccAn));
        i *= 2;
}
```

  - Newton's method in DA doubles the number of correct orders in each iteration