tf.contrib.integrate.odeint

```
odeint(
    func,
    y0,
    t,
    rtol=1e-06,
    atol=1e-12,
    method=None,
    options=None,
    full_output=False,
    name=None
)
```

Defined in `tensorflow/contrib/integrate/python/ops/odes.py`.

See the guide: Integrate (contrib) > Ops

Integrate a system of ordinary differential equations.

Solves the initial value problem for a non-stiff system of first order ODEs:

```
dy/dt = func(y, t), y(t[0]) = y0
```

where y is a Tensor of any shape.

For example:

```
# solve `dy/dt = -y`, corresponding to exponential decay
tf.contrib.integrate.odeint(lambda y, _: -y, 1.0, [0, 1, 2])
=> [1, exp(-1), exp(-2)]
```

Output dtypes and numerical precision are based on the dtypes of the inputs `y0` and `t`.

Currently, implements 5th order Runge-Kutta with adaptive step size control and dense output, using the Dormand-Prince method. Similar to the 'dopri5' method of `scipy.integrate.ode` and MATLAB's `ode45`.

Based on: Shampine, Lawrence F. (1986), "Some Practical Runge-Kutta Formulas", Mathematics of Computation, American Mathematical Society, 46 (173): 135-150, doi:10.2307/2008219

Args:

- `func`: Function that maps a Tensor holding the state `y` and a scalar Tensor `t` into a Tensor of state derivatives with respect to time.
- `y0`: N-D Tensor giving starting value of `y` at time point `t[0]`. May have any floating point or complex dtype.
- `t`: 1-D Tensor holding a sequence of time points for which to solve for `y`. The initial time point should be the first element of this sequence, and each time must be larger than the previous time. May have any floating point dtype. If not provided as a Tensor, converted to a Tensor with float64 dtype.
- `rtol`: optional float64 Tensor specifying an upper bound on relative error, per element of `y`.
- `atol`: optional float64 Tensor specifying an upper bound on absolute error, per element of `y`.

- `method`: optional string indicating the integration method to use. Currently, the only valid option is `'dopri5'`.
- `options`: optional dict of configuring options for the indicated integration method. Can only be provided if a `method` is explicitly set. For
  `'dopri5'`, valid options include:
  - first_step: an initial guess for the size of the first integration (current default: 1.0, but may later be changed to use heuristics based on the gradient).
  - safety: safety factor for adaptive step control, generally a constant in the range 0.8-1 (default: 0.9).
  - ifactor: maximum factor by which the adaptive step may be increased (default: 10.0).
  - dfactor: maximum factor by which the adpative step may be decreased (default: 0.2).
  - max_num_steps: integer maximum number of integrate steps between time points in `t` (default: 1000).
- `full_output`: optional boolean. If True, `odeint` returns a tuple `(y, info_dict)` describing the integration process.
- `name`: Optional name for this operation.

Returns:

- `y`: (N+1)-D tensor, where the first dimension corresponds to different time points. Contains the solved value of y for each desired time point in `t`, with the initial value `y0` being the first element along the first dimension.
- `info_dict`: only if `full_output == True`. A dict with the following values:
  - num_func_evals: integer Tensor counting the number of function evaluations.
  - integrate_points: 1D float64 Tensor with the upper bound of each integration time step.
  - error_ratio: 1D float Tensor with the estimated ratio of the integration error to the error tolerance at each integration step. An ratio greater than 1 corresponds to rejected steps.

Raises:

- `ValueError`: if an invalid `method` is provided.
- `TypeError`: if `options` is supplied without `method`, or if `t` or `y0` has an invalid dtype.
