

hsqd-vec

Claude Heiland-Allen

2011–2015

Contents

1	.gitignore	2
2	LICENSE	2
3	Numeric/QD/Vec.hs	3
4	qd-vec.cabal	4
5	Setup.hs	5

1 .gitignore

dist

2 LICENSE

Copyright (c) 2011,2012, Claude Heiland-Allen

All rights reserved.

5 Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

- 10 * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
 - 15 * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
with the distribution.
 - * Neither the name of Claude Heiland-Allen nor the names of other
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.
- 20 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
25 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
30 OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3 Numeric/QD/Vec.hs

```

{-# LANGUAGE FlexibleInstances , TypeFamilies , TypeSynonymInstances #-}
{- |
Module      :  Numeric.QD.Vec
Copyright   :  (c) Claude Heiland-Allen 2011, 2012
5  License   :  BSD3

Maintainer  :  claud@mathr.co.uk
Stability   :  provisional
Portability :  TypeFamilies , TypeSynonymInstances , FlexibleInstance
10
Vec instances for QD types.
-}
module Numeric.QD.Vec where
import Data.Vec (NearZero(..), PackedVec(..), Vec2, Vec3, Vec4, (:.)(:.))
15 import Numeric.QD (DoubleDouble, QuadDouble)

instance NearZero DoubleDouble where nearZero x = not (abs x > (1e-28))

instance NearZero QuadDouble where nearZero x = not (abs x > (1e-60))
20
instance PackedVec (Vec2 DoubleDouble) where
  data Packed (Vec2 DoubleDouble) = Vec2DD {-#UNPACK#-} !DoubleDouble {-#UNPACK#-}
    ↳ #-} !DoubleDouble
  pack (a:.b:.) = Vec2DD a b
  unpack (Vec2DD a b) = a:.b:.(.)
25 {-# INLINE pack #-}
   {-# INLINE unpack #-}

instance PackedVec (Vec3 DoubleDouble) where
  data Packed (Vec3 DoubleDouble) = Vec3DD {-#UNPACK#-} !DoubleDouble {-#UNPACK#-}
    ↳ #-} !DoubleDouble {-#UNPACK#-} !DoubleDouble
30 pack (a:.b:.c:.) = Vec3DD a b c
   unpack (Vec3DD a b c) = a:.b:.c:.(.)
   {-# INLINE pack #-}
   {-# INLINE unpack #-}

35 instance PackedVec (Vec4 DoubleDouble) where
  data Packed (Vec4 DoubleDouble) = Vec4DD {-#UNPACK#-} !DoubleDouble {-#UNPACK#-}
    ↳ #-} !DoubleDouble {-#UNPACK#-} !DoubleDouble {-#UNPACK#-} !DoubleDouble
   pack (a:.b:.c:.d:.) = Vec4DD a b c d
   unpack (Vec4DD a b c d) = a:.b:.c:.d:.(.)
40 {-# INLINE pack #-}
   {-# INLINE unpack #-}

type Vec2DD = Packed (Vec2 DoubleDouble)
type Vec3DD = Packed (Vec3 DoubleDouble)
type Vec4DD = Packed (Vec4 DoubleDouble)
45
type Mat22DD = Vec2 (Vec2DD)
type Mat23DD = Vec2 (Vec3DD)
type Mat24DD = Vec2 (Vec4DD)

50 type Mat32DD = Vec3 (Vec2DD)
type Mat33DD = Vec3 (Vec3DD)
type Mat34DD = Vec3 (Vec4DD)

```

```

type Mat42DD = Vec4 (Vec2DD)
55 type Mat43DD = Vec4 (Vec3DD)
type Mat44DD = Vec4 (Vec4DD)

instance PackedVec (Vec2 QuadDouble) where
  data Packed (Vec2 QuadDouble) = Vec2QD {-#UNPACK#-} !QuadDouble {-#UNPACK#-} !↯
    ↳ QuadDouble
60 pack (a:.b:.( )) = Vec2QD a b
  unpack (Vec2QD a b) = a:.b:.( )
  {-# INLINE pack #-}
  {-# INLINE unpack #-}

65 instance PackedVec (Vec3 QuadDouble) where
  data Packed (Vec3 QuadDouble) = Vec3QD {-#UNPACK#-} !QuadDouble {-#UNPACK#-} !↯
    ↳ QuadDouble {-#UNPACK#-} !QuadDouble
  pack (a:.b:.c:.( )) = Vec3QD a b c
  unpack (Vec3QD a b c) = a:.b:.c:.( )
  {-# INLINE pack #-}
70 {-# INLINE unpack #-}

instance PackedVec (Vec4 QuadDouble) where
  data Packed (Vec4 QuadDouble) = Vec4QD {-#UNPACK#-} !QuadDouble {-#UNPACK#-} !↯
    ↳ QuadDouble {-#UNPACK#-} !QuadDouble {-#UNPACK#-} !QuadDouble
  pack (a:.b:.c:.d:.( )) = Vec4QD a b c d
75 unpack (Vec4QD a b c d) = a:.b:.c:.d:.( )
  {-# INLINE pack #-}
  {-# INLINE unpack #-}

type Vec2QD = Packed (Vec2 QuadDouble)
80 type Vec3QD = Packed (Vec3 QuadDouble)
type Vec4QD = Packed (Vec4 QuadDouble)

type Mat22QD = Vec2 (Vec2QD)
type Mat23QD = Vec2 (Vec3QD)
85 type Mat24QD = Vec2 (Vec4QD)

type Mat32QD = Vec3 (Vec2QD)
type Mat33QD = Vec3 (Vec3QD)
type Mat34QD = Vec3 (Vec4QD)
90 type Mat42QD = Vec4 (Vec2QD)
type Mat43QD = Vec4 (Vec3QD)
type Mat44QD = Vec4 (Vec4QD)

```

4 qd-vec.cabal

```

Name:                qd-vec
Version:             1.1
Synopsis:            'Vec' instances for 'qd' types
Description:
5   Instances of Vec's NearZero and Packed for qd's types.

License:             BSD3
License-file:        LICENSE
Author:              Claude Heiland-Allen
10 Maintainer:       claud@mathr.co.uk

```

```
Category:      Math
Build-type:    Simple
cabal-version: >=1.8
```

```
15  Library
    Build-depends: base >= 3 && < 6, qd >= 1 && < 2, Vec >= 0.9.9 && < 2
    Exposed-modules: Numeric.QD.Vec
    GHC-options:   -Wall -fno-excess-precision
    GHC-prof-options: -auto-all -caf-all
```

```
20  source-repository head
    type:      git
    location:   http://code.mathr.co.uk/hsqd-vec.git
```

```
25  source-repository this
    type:      git
    location:   http://code.mathr.co.uk/hsqd-vec.git
    tag:       v1.1
```

5 Setup.hs

```
import Distribution.Simple
main = defaultMain
```