$$f(x) = \sum F_i b_i(x)$$

f(x) = \sum F_i b_i(x)

$$\int f(x) w(x) dx \approx \int f(x) w^h(x) dx = \sum c_i \int f(x) b_i(x) dx$$

\int f(x) w(x) dx \approx \int f(x) w^h(x) dx = \sum c_i \int f(x) b_i(x) dx

$$\int w(x) f(x) dx \approx \int w^h(x) f^h(x) dx = \sum_i \sum_j c_i F_j \int b_i(x) b_j(x) dx$$

\int w(x) f(x) dx \approx \int w^h(x) f^h(x) dx = \sum_i \sum_j c_i F_j
\int b_i(x) b_j(x) dx

Nodal basis function

$$b_i(x_k) = \delta_{ij}$$

b_i (x_k) = \delta_{ij}

Then $$F_k = f(x_k)$$

F_k = f(x_k)

Non-nodal basis function: Need to compute b_k by projection using the Gram matrix; recall the initial condition, need to do projection with the mass matrix.

$$\left\{ F_k \right\} = \boldsymbol \Gamma \left\{ < b_j , f > \right\}$$

\left\{ F_k \right\} = \boldsymbol \Gamma \left\{ <b_j , f> \right\}

$$\boldsymbol{\Gamma} = \left[< b_i , b_j > \right]$$

\boldsymbol \Gamma = \left[<b_i , b_j> \right]

Q: How does fenics do the above operation to compute the right-hand side ?

Note: It is important to answer the above question, since then you would know how to set up the initial condition.

Guess of how fenics works:

Assume that you would discretize f(x) using the same basis functions like that of the "mesh" (triangular, etc.)

basis function for the mesh would be b_i (x) such that

$$f(x) = \sum F_i b_i(x)$$

f(x) = \sum F_i b_i(x)

$$\int f(x) w(x) dx \approx \int f(x) w^h(x) dx = \sum c_i \int f(x) b_i(x) dx$$

\int f(x) w(x) dx \approx \int f(x) w^h(x) dx = \sum c_i \int f(x) b_i(x) dx

$$\int w(x)f(x)dx \approx \int w^h(x)f^h(x)dx = \sum_i \sum_j c_i F_j \int b_i(x)b_j(x)dx$$

\int w(x) f(x) dx \approx \int w^h(x) f^h(x) dx = \sum_i \sum_j c_i F_j \int b_i(x) b_j(x) dx

assume nodal basis function so that the coeff. F_j is just the value of the function f(x) at x_j.

Nodal basis function

$$b_i(x_k) = \delta_{ij}$$

b_i (x_k) = \delta_{ij}

Then

$$F_k = f(x_k)$$

F_k = f(x_k)

So fenics needs to compute the coeff of the Gram matrix, i.e.,

$$\Gamma_{ij} =< b_i, b_j >= \int b_i(x)b_j(x)dx$$

\Gamma_{ij} = <b_i , b_j> = \int b_i(x) b_j(x) dx

so fenics may have a code that computes the above coefficients using Gauss quad (question to answer)

even at this stage (where you had solve the Poisson's equation), there is a lot that is hidden behind the code.

cursor motion mode

map   [up arrow] = k

insert mode

map!   [up arrow] = <esc>k